ELSEVIER

Theoretical Computer Science 202 (1998) 55–125

Theoretical
Computer Science

Fundamental Study

Programming in metric temporal logic[1]

Christoph Brzoska*

Institut für Logik, Komplexität und Deduktionssysteme, Universität Karlsruhe (TH), Postfach 69 80, D-76128 Karlsruhe, Germany

Received August 1996; revised June 1997
Communicated by M. Nivat

Abstract

We present a fragment of metric temporal logic called *bounded universal Horn formulae* as a theoretical basis for temporal reasoning in logic programming. We characterize its semantics in terms of fixed points and canonical models, and present an efficient proof method as operational semantics based on SLD-resolution with constraints. Although the complexity of real-time logics is very high in general – the validity problem for most of them is Π_1^1-complete already for propositional fragments in case of dense time structures – we show that the class of bounded universal Horn formulae admits complete and efficient proof methods exploiting uniform proofs and linear time complexity of basic steps of the proof method. The results obtained heavily rely on the fragment investigated and make it necessary to establish some basic results like compactness and approximation of the least model by at most ω-steps of the corresponding fixed point operator directly without recourse to standard methods (in dense case). The fragment itself is sufficiently expressive for a variety of applications ranging from real-time systems, temporal (deductive) data bases, and sequence evaluation purposes. We show that the fragment is the greatest of the metric temporal logic – in discrete and dense case – having the properties classically desired for logic programming languages. © 1998—Elsevier Science B.V. All rights reserved

Keywords: Logic programming; Temporal logic; Temporal data bases; Theorem proving

Contents

* E-mail: brzoska@ira.uka.de.

[1] This research has been supported by the Deutsche Forschungsgemeinschaft, SFB 314 (S 2), and ESPRIT program of the European community, working group "Logic & Change" and "Compulog" net work of excellence.

PII S0304-3975(97)00139-4

1. Introduction

Logic programming based on Horn formulae has been established as one of the main approaches to declarative programming. Originally motivated by computer linguistic applications [33], theorem proving experience [55, 67], and methodological considerations [19, 93] it has emancipated and developed to a general purpose programming paradigm. Several extensions have been proposed with some of them forming a programming paradigm on its own like *constraint logic programming* [59], *concurrent constraint logic programming* [96], *functional logic programming* [35, 52], *disjunctive logic programming* [76], or *intuitionistic logic programming* [49, 83, 79, 80].

Also extensions towards formalisms allowing explicit reasoning about time and temporal dependencies have been presented, some of them based on a temporal logic [46, 48, 2, 15], and some within the constraint logic programming paradigm using special temporal theories [56, 57, 44]. Although time and its handling is central for many applications in computer science and artificial intelligence no widely accepted basis for logic programming with temporal reasoning capabilities has been approved so far.

We propagate in this paper a class of *bounded universal* (*modality*) *Horn formulae* defined in the following as such a basis. This class admits an efficient operationalization comparable to classical logic programming languages like PROLOG, semantical characterizations in terms of least and greatest models, least and greatest fixed points of a suitable consequence operator, and the integration of negation as failure using an efficient specialization of constructive negation [98]. In the first part of the paper, we concentrate on discrete time structures – integers and natural numbers; in the second, on dense time – rational and real flow of time – and sketch the integration of negation as failure.

Temporal logics based on discrete time models – i.e., using integers, natural numbers, or (discrete) trees – have been extensively used for the specification and verification of (concurrent) programs [72, 77, 78, 39] since the first proposals in [91, 71, 10]. They seem to be adequate for the description of *synchronous* systems, where all parts are driven by a common clock. However, for the step-wise refinements and composition of specifications it has been argued in [73] already that specifications have to be invariant under stuttering, which restricts the usage of the "next" operator in the usual discrete time temporal logic formalisms. Alternatively, invariance under stuttering can be achieved by introducing a rational time semantics as it has been proposed in [66]. Dense time seems to be also the more natural choice in many AI and data base applications [101] and allows furthermore to model *asynchronous systems* appropriately, where all parts of the system are not necessarily driven by a common clock.

The complexity of real-time logics over dense time structures, however, is very high. It has been shown in [7] that for each (propositional) real-time logic over dense time allowing addition by constants the validity problem is Π_1^1-hard and, as a consequence, for the logics proposed in [62, 69, 90].

Interestingly, bounded universal Horn formulae over dense time admit complete proof methods of the same complexity as their discrete counterparts. In this paper we present a proof method for discrete time and two calculi for dense time bounded universal Horn formulae, prove their soundness and completeness, some complexity results for both discrete and dense time, and sketch an efficient specialization of constructive negation for one of the proof methods.

The proof methods presented in the paper have several interesting properties. Firstly, each step of the calculi can be performed in linear time. Secondly, the calculi presented are optimal with respect to the derivations needed to prove bounded universal properties, i.e. properties expressible by $\Box_I A$ for a finite interval I, $A \mathcal{S} B$ or $A \mathcal{U} B$, since they are able to exploit uniform proofs for A. Thirdly, the integration of constructive negation is fully compatible with the second proof method proposed and allows thereby to utilize uniform proofs for queries built up over bounded universal temporal operators and all logical connectives of the classical logic with negation interpreted as negation as failure. These properties predestinate the class and the proof methods for temporal databases, knowledge base, and AI applications relying on efficient temporal reasoning capabilities.

The results obtained heavily depend – especially, in the dense case – on the fragment investigated and make it necessary to establish some basic results like compactness and approximation of the least model by at most ω-steps of the corresponding fixed point operator explicitly without recourse to standard methods in the dense case. Unlike classical approaches to theorem proving in temporal logics we do not use recursive characterizations of the operators of temporal logic but use (functional) translations into classical logic with fixed interpretations of symbols and relations modeling time – linear inequalities over integers and rationals. Such translations and proof methods based upon have been introduced recently in the context of modal logics (cf. [106, 86, 37, 11, 85, 43]). In contrast to those, however, in the temporal case quantifiers introduced by the reinterpretation in classical logic can not be eliminated by Skolemization due to the fixed interpretation of the new symbols and the time domains. We use instead quantifier elimination methods for existential and bounded universal quantifiers over linear inequalities. For reasons of efficiency, we do not utilize general quantifier elimination algorithms – e.g., those for Presburger and real arithmetic – but develop special algorithms based on Fourier's algorithms for solving linear inequalities. Although Fourier's method is exponential in general, elimination of existential and bounded universal quantifiers, which have been introduced by the translations and which have to be treated appropriately during derivations, can be performed in linear time by the specialized algorithms. As a byproduct, we obtain thereby a new quantifier elimination method for bounded universal quantifiers over systems of linear inequalities over the rationals and the reals, and over the integers for a restricted class of inequalities called *tree constraint systems* defined in this paper.

Extensions of the class of bounded universal Horn formulae either leads to highly incomplete logics – to Π_1^1-complete ones, lost of least and greatest model and fixed point characterizations, or the ability to exploit uniform proofs. We argue that the class is sufficiently expressive for a variety of applications ranging from temporal (deductive) data bases (Fig. 1), (image) sequence evaluation purposes [27] to real-time systems (Fig. 2). It has been already taken as basis for the development of the temporal logic programming system *Limette* [28].

The paper is organized as follows. In Sections 2 and 3 we introduce the temporal logic underlying our work and the class of temporal Horn formulae – *bounded universal modality Horn formulae* – being investigated subsequently. Sections 4 and 5 are devoted to semantical characterizations in terms of fixed points and canonical models. Sections 6–11, this is the most important part of the paper, are concerned with the operational semantics for bounded universal modality Horn formulae. Several proof method based on SLD-resolution with constraints – linear inequalities over the integers (respectively, the rational numbers) – are presented and several complexity results are proven, for example, the incremental complexity of each step of the calculi is shown to be linear. In general, however, already checking linear inequalities over the integers – in case of discrete time structures – for satisfiability is well-known as being NP-complete. The methods exploits uniform proofs for the universally quantified goals to be proven, contrary to standard proof methods in temporal logics. The remaining Sections 12–15 discuss integration of the negation as failure rule, extensions of the class of temporal Horn formulae under consideration, anchored flows of time, and related works.

This paper is based on [21–25], which contain parts of this work in a preliminary version.

$\Box_{[-60,-37]}$ empl(john, 60 000, toys)
$\Box_{[-36,-31]}$ empl(john, 60 000, shoes)
$\Box_{[-30,-25]}$ empl(john, 70 000, shoes)
$\Box_{[-24,-13]}$ empl(john, 80 000, shoes)
$\Box_{[-12,-1]}$ empl(john, 90 000, clothing)

⋮

$\Box_{[-72,-61]}$ salesman_in(john, toys)
$\Box_{[-60,-37]}$ manager_of(john, toys)
$\Box_{[-36,-25]}$ manager_of(leu, toys)
$\Box_{[-36,-13]}$ manager_of(john, shoes)
$\Box_{[-12,-1]}$ manager_of(john, clothing)

⋮

$\Box$ (manager(X) $\leftarrow$ manager_of(X, D))
$\Box$ (salesman(X) $\leftarrow$ salesman_in(X, Y))
$\Box$ (salary(X, S) $\leftarrow$ empl(X, S, Y))

Query: Is there a manager who have had a salary of at least 70000 $ for 2 years within the last 5 years?

$$\Diamond_{[-60,-1]}\Box_{[0,23]}\ (\mathrm{manager(X)} \wedge \exists\, \mathrm{S(salary(X,S)} \wedge \mathrm{S} > 70000)).$$

Fig. 1. A temporal data base represented by a temporal logic program.

$$\Box\ (\mathrm{do\_backup(X)}\ \leftarrow \Box_{[-24,-1]}\, \mathrm{not\ backup(X)})$$

Fig. 2. A temporal logic program controlling backups.

2. Logic

We base our work on *metric temporal logic* [68–70]. Their formulae are built up with the usual logical connectives and following temporal operators: $\Box_I$ (always), $\Diamond_I$ (sometime) within the interval I, $\mathcal{S}$ (A has always been true, since B was true), $\mathcal{U}$ (A will be always true in the future, until B will be true), where $I=[c^-,c^+]$ with $c^-,c^+ \in \mathbb{Z}\cup\{-\infty,\infty\}$ in the discrete case, $I\in\{[c^-,c^+],(c^-,c^+],[c^-,c^+),(c^-,c^+)\}$ with $c^-,c^+\in\mathbb{Q}\cup\{-\infty,\infty\}$ in the dense case. Metric temporal logic generalizes linear temporal logic as it has been introduced in [91]. The idea to use temporal operators to express metric temporal constraints is very natural and has been already mentioned in [94, 29].

The binary operators $\mathscr{S}$ and $\mathscr{U}$ have been introduced by H. Kamp in his thesis in 1968. They are expressive complete for the integer and real flow of time. Their metric versions – $\mathscr{S}_I$ and $\mathscr{U}_I$ – specify an interval, within which the second argument has to be true. $\Box_I$ and $\Diamond_I$ generalize classical temporal operators $\Box$ (always) and $\Diamond$ (sometime) originating in modal logics.

We restrict our attention to temporal logics with function symbols interpreted time-independently, i.e., following the modal logic terminology to *rigid function symbols*, and predicate symbols with denotations varying with time. They are called *flexible* according to modal logic conventions [58]. We use standard notations (cf. [75, 59, 61]) which are briefly surveyed in Appendix A.

Definition 2.1 (*MTL-Σ-Structure*). Let Σ be a signature. We call a Kripke Σ-structure $(\mathscr{D}, \mathscr{T}, t_0, <, \mathscr{I})$ *discrete (MTL-)Σ-structure* if

- $(\mathscr{T}, t_0, <) \simeq (\mathbb{Z}, 0, <)$, this is, the set of time points is isomorphic to the integers with $0 \in \mathbb{Z}$ as the first time point, and $<$ on $\mathbb{Z}$ as the before-relation on $\mathscr{T}$,
- $\mathscr{D} = \bigcup_{s \in S} \mathscr{D}_s$ with $\mathscr{D}_s \neq \emptyset$ for every $s \in S$,
- The *interpretation* $\mathscr{I}$ assigns to each function symbol $f : s_1 \ldots s_n \to s$ in Σ a mapping $f^{\mathscr{M}}$ from $\mathscr{D}_{s_1} \times \cdots \times \mathscr{D}_{s_n}$ to $\mathscr{D}_s$ and to each predicate symbol $p : s_1 \ldots s_n$ in Σ and each time point t in $\mathscr{T}$ a relation $p_t^{\mathscr{M}} \subseteq \mathscr{D}_{s_1} \times \cdots \times \mathscr{D}_{s_n}$.

A MTL-Σ-structure as defined above except $(\mathscr{T}, t_0, <) \simeq (\mathbb{Q}, 0, <)$ is called *dense (MTL-)Σ-structure*.

Validity in MTL-structures is defined as usual in temporal logics.

Definition 2.2. The *Validity* of a formula A in a (MTL-Σ-)structure $\mathscr{M}$ at time t under a variable assignment α, denoted by $(\mathscr{M}, \alpha) \models_t A$, is defined by

1. $(\mathscr{M}, \alpha) \models_t p(r_1, \ldots, r_n)$ iff $p_t^{\mathscr{M}}(\overline{\alpha}(r_1), \ldots, \overline{\alpha}(r_n))$ holds in $\mathscr{M}$, for every predicate symbol $p : s_1 \ldots s_n$ in Σ and terms $r_i \in \mathscr{T}_\Sigma(\mathscr{V})s_j$, $i = 1, \ldots, n$,
2. $(\mathscr{M}, \alpha) \models_t \Box_I A$ iff for all $t' \in I$, $(\mathscr{M}, \alpha) \models_{t+t'} A$,
3. $\mathscr{M}, \alpha) \models_t \Diamond_I A$ iff for some $t' \in I$, $(\mathscr{M}, \alpha) \models_{t+t'} A$,
4. $(\mathscr{M}, \alpha) \models_t A \; \mathscr{S}_I \; B$ iff there is a $t' < 0$ and $t' \in I$ such that $(\mathscr{M}, \alpha) \models_{t+t'} B$ and for all t'' with $t + t' < t'' < t$, $(\mathscr{M}, \alpha) \models_{t''} A$,
5. $(\mathscr{M}, \alpha) \models_t A \; \mathscr{U}_I \; B$ iff there is a $t' > 0$ and $t' \in I$ such that $(\mathscr{M}, \alpha) \models_{t+t'} B$ and for all t'' with $t < t'' < t + t'$, $(\mathscr{M}, \alpha) \models_{t''} A$,

where $t' \in [a, b]$ iff $a \leqslant t' \leqslant b$, $t' \in (a, b]$ iff $a < t' \leqslant b$, $t' \in [a, b)$ iff $a \leqslant t' < b$, $t' \in (a, b)$ iff $a < t' < b$, and $-\infty < c < \infty$ for all $c \in \mathbb{Z}$ (respectively, $c \in \mathbb{Q}$).

The remaining cases, $A \wedge B$, $A \vee B$, $A \to B$, $\neg A$, $\forall x A$, and $\exists x A$, are defined as usual. A formula A *is valid* (*under* α) *in* $\mathscr{M}$ iff $(\mathscr{M}, \alpha) \models_{t_0} A$; A *is valid* (*in* $\mathscr{M}$) iff $(\mathscr{M}, \alpha) \models_{t_0} A$ for all variable assignments $\alpha : \mathscr{V} \to \mathscr{M}$. The *satisfiability* and *logical consequence*, denoted by $\models$, are defined in the standard way. Classical temporal operators and their metric versions are defined in Fig. 3.

$\Box A \leftrightarrow \Box_{(-\infty,\infty)} A$	$\Diamond A \leftrightarrow \Diamond_{(-\infty,\infty)} A$	
$\Box_+ A \leftrightarrow \Box_{[0,\infty)} A$	$\Box_- A \leftrightarrow \Box_{(-\infty,0]} A$	
$\Diamond_+ A \leftrightarrow \Diamond_{[0,\infty)} A$	$\Diamond_- A \leftrightarrow \Diamond_{(-\infty,0]} A$	
$A\,\mathcal{S}_c\,B \leftrightarrow A\,\mathcal{S}_{[c,0)}\,B$	$A\,\mathcal{U}_c\,B \leftrightarrow A\,\mathcal{U}_{(0,c]}\,B$	
$A\,\mathcal{S}\,B \leftrightarrow A\,\mathcal{S}_{(-\infty,0)}\,B$	$A\,\mathcal{U}\,B \leftrightarrow A\,\mathcal{U}_{(0,\infty)}\,B$	
$\circ A \leftrightarrow \Box_{[1,1]} A$	$\bullet A \leftrightarrow \Box_{[-1,-1]} A$	
$\circ A \leftrightarrow \Diamond_{[1,1]} A$	$\bullet A \leftrightarrow \Diamond_{[-1,-1]} A$	
$\Box_{\leqslant c} A \leftrightarrow \Box_{[0,c]} A$	$\Box_{<c} A \leftrightarrow \Box_{[0,c)} A$	$c \geqslant 0$
$\Diamond_{\leqslant c} A \leftrightarrow \Diamond_{[0,c]} A$	$\Diamond_{<c} A \leftrightarrow \Diamond_{[0,c)} A$	$c \geqslant 0$
$\Box_{\geqslant c} A \leftrightarrow \Box_{[c,\infty)} A$	$\Box_{>c} A \leftrightarrow \Box_{(c,\infty)} A$	$c \geqslant 0$
$\Diamond_{\geqslant c} A \leftrightarrow \Diamond_{[c,\infty)} A$	$\Diamond_{>c} A \leftrightarrow \Diamond_{(c,\infty)} A$	$c \geqslant 0$
$\Box_{\geqslant c} A \leftrightarrow \Box_{[c,0]} A$	$\Box_{>c} A \leftrightarrow \Box_{(c,0]} A$	$c \leqslant 0$
$\Diamond_{\geqslant c} A \leftrightarrow \Diamond_{[c,0]} A$	$\Diamond_{>c} A \leftrightarrow \Diamond_{(c,0]} A$	$c \leqslant 0$
$\Box_{\leqslant c} A \leftrightarrow \Box_{(-\infty,c]} A$	$\Box_{<c} A \leftrightarrow \Box_{(-\infty,c)} A$	$c \leqslant 0$
$\Diamond_{\leqslant c} A \leftrightarrow \Diamond_{(-\infty,c]} A$	$\Diamond_{<c} A \leftrightarrow \Diamond_{(-\infty,c)} A$	$c \leqslant 0$
$\Box_c A \leftrightarrow \Box_{\leqslant c} A$	$\Diamond_c A \leftrightarrow \Diamond_{\leqslant c} A$	$c \geqslant 0$
$\Diamond_c A \leftrightarrow \Diamond_{\geqslant c} A$	$\Box_c A \leftrightarrow \Box_{\geqslant c} A$	$c \leqslant 0$

Fig. 3. Derived temporal operators.

3. Bounded universal (modality) goals

We focus on temporal Horn formulae containing $\Diamond_I$, $\mathcal{S}_I$, $\mathcal{U}_I$, and $\Box_{I'}$ operators in goals and bodies, where I may be a bounded or unbounded interval over the integers (respectively, over the rationals in the dense case) and I' a bounded interval – over the integers (respectively, over the rationals).

Formally, the goals are called *bounded universal* (*modality*) *goals* and are defined by

$$G ::= \varepsilon \mid A \mid \Diamond_I\, G \mid G \wedge G \mid \Box_{I'} G \mid G\,\mathcal{S}_I\, G \mid G\,\mathcal{U}_I\, G,$$

the Horn formulae called *bounded universal* (*modality*) *Horn formulae* by

$$D ::= A \mid \Box_I D \mid D \leftarrow G,$$

where I denotes an interval and I' an with bounds in $\mathbb{Z}$ (respectively, $\mathbb{Q}$ in the dense case), this is $I' = |^- c^-, c^+ |^+$ with $c^-, c^+ \in \mathbb{Z}$ (respectively, $c^-, c^+ \in \mathbb{Q}$) and $|^- \in \{(,[\}$, $|^+ \in \{),]\}$. ε denotes the empty goal and A ranges over atoms.

In the discrete case bounded universal goals and Horn formulae can be also defined using $\circ$, $\bullet$, $\Diamond_c$, and $\Box_c$ operators as basis. Goals are then defined by

$$G ::= \varepsilon \mid A \mid \circ G \mid \bullet G \mid \Diamond_c\, G \mid \Box_i G \mid G\,\mathcal{S}_c\, G \mid G\,\mathcal{U}_c\, G \mid G \wedge G$$

and Horn formulae by

$$D ::= A \mid \circ D \mid \bullet D \mid \Box_c D \mid D \leftarrow G$$

with $c \in \mathbb{Z} \cup \{-\infty, \infty\}$, and $i \in \mathbb{Z}$.

The class is sufficiently expressive for a variety of applications ranging from temporal (deductive) data bases (Fig. 1), and (image) sequence evaluation [27] to real-time systems (cf. Fig. 2 and the examples listed below).

Example 3.1. Time outs in communication protocols:

$$\Box(\Box_+ \text{served\_in\_time}(A,M) \leftarrow \Diamond_{-t}\, \text{send}(A,M) \wedge \text{acknowledge}(A,M)),$$

$$\Box(\Box_+ \text{time\_out}(A,M) \leftarrow \textit{not}\ \Diamond_{-t}\, \text{served\_in\_time}(A,M)),$$

send(A,M) models sending message M at address A, acknowledge(A,M) models getting an acknowledgment for sent message M at A; served_in_time(A,M), time_out(A,M) specifies serving request in time (respectively, not in time).

Example 3.2. Robot motion planning: Assume we have for each action a_i of the robot a minimal execution time c_i^- and a maximal execution time c_i^+, and for each pair a, b of actions a minimal and a maximal reconfiguration time ab^- and ab^+. We model the minimal and maximal execution time for each a_i by formulae

$$\Box(\Box_{[c_i^-, c_i^+]} \mathit{end}(a_i) \leftarrow \mathit{start}(a_i))$$

and the reconfiguration time for each pair of actions a and b by Horn formulae

$$\Box(\Box_{[ab^-, ab^+]} \mathit{start}(b) \leftarrow \mathit{end}(a)).$$

The requirement of performing some action with priority if their execution can be performed within 60 seconds can be specified using a bounded universal Horn formula, namely by

$$\Box(\text{do\_next}(X) \leftarrow \text{request\_for}(X) \wedge \text{priority}(X) \wedge \Diamond_{60}\, \text{start}(X)).$$

4. Translation into first-order logic

Following the approach of functional translation into first-order logic pioneered by [106, 86] and others for modal logics, we will translate formulae of the temporal logic under investigation into first-order logic with fixed interpretation of some symbols modeling the flow of time. Such translations from modal into classical logic are also of independed interest in the context of the *correspondence theory* with respect to model-theoretic and axiomatization questions [104, 103]. In the following, we use those functional translations both to derive efficient proof methods for the class of bounded universal Horn formulae and to characterize their semantics in model-theoretic terms exploiting the bijective correspondence between temporal and first-order structures over a fixed algebra pointed out in [21].

The idea is to add an additional argument to each predicate and to express the temporal relations which are expressed by temporal operators in temporal logic by formulae of classical logic. More precisely, temporal Σ-formulae are translated into formulae over an enriched signature $\Pi(\Sigma)=(\Pi(S),\Pi(F),\Pi(P))$ with

$$\begin{aligned}
&\Pi(S)=S\uplus\{\mathit{time}\},\\
&\Pi(F)=F\uplus\{0:\to\mathit{time}, +:\mathit{time}\ \mathit{time}\to\mathit{time}\},\ \text{and}\\
&\Pi(P)=\{p:\mathit{time}\ s_1\ldots s_n \mid p:s_1\ldots s_n\in P\}\uplus\\
&\qquad\{=:\mathit{time}\ \mathit{time},\ \leqslant:\mathit{time}\ \mathit{time},\ <:\mathit{time}\ \mathit{time}\},
\end{aligned}$$

where $\uplus$ denotes the disjoint union of sets. The translation itself is defined[2] by

$$\begin{aligned}
&\Pi(A)=\pi(A,0,\emptyset)\\
&\pi(p(\mathbf{r}),t,C)=p(t,\mathbf{r})\\
&\pi(\Box_I A,t,C)=\forall x(\{x\in I\}\to\pi(A,t+x,\{x\in I\}\cup C))\\
&\pi(\Diamond_I A,t,C)=\exists x(\{x\in I\}\wedge\pi(A,t+x,\{x\in I\}\cup C))\\
&\pi(A\,\mathcal{S}_I\,B,t,C)=\exists y(\{y<0,y\in I\}\wedge\pi(B,t+y,\{y<0,y\in I\}\cup C)\\
&\qquad\qquad\wedge\forall y'(\{y<y'<0\}\to\pi(A,t+y',\{y<y'<0\}\cup C)))\\
&\pi(A\,\mathcal{U}_I\,B,t,C)=\exists y(\{y>0,y\in I\}\wedge\pi(B,t+y,\{y>0,y\in I\}\cup C)\\
&\qquad\qquad\wedge\forall y'(\{0<y'<y\}\to\pi(A,t+y',\{0<y'<y\}\cup C)))\\
&\pi(A\leftarrow B,t,C)=\pi(A,t,C)\leftarrow\pi(B,t,C)\\
&\pi(A\wedge B,t,C)=\pi(A,t,C)\wedge\pi(B,t,C)
\end{aligned}$$

where $y\in I$ stands for $i^-\leqslant y\leqslant i^+$ if $I=[i^-,i^+]$, $i^-<y\leqslant i^+$ if $I=(i^-,i^+]$, $i^-\leqslant y<i^+$ if $I=[i^-,i^+)$, $i^-<y<i^+$ if $I=(i^-,i^+)$, and $\mathbf{r}$ denotes a tuple of terms $r_1,\ldots,r_n$.

Remarks. (i) $x\leqslant\infty$ (respectively, $-\infty\leqslant x$) represents an empty constraint, i.e., $x\leqslant\infty$ (respectively, $-\infty\leqslant x$) is always true.

(ii) For notational convenience, we often drop the last argument of π and write π as a binary function if the last argument of π is not important or it is clear from the context (i.e., we write $\pi(A,t)$ instead of $\pi(A,t,C)$).

The formulae translated are interpreted in classical first-order structures with fixed interpretation of *time*, 0, +, = and $\leqslant$ as integers $\mathbb{Z}$, $0\in\mathbb{Z}$, addition, equality, and inequality over $\mathbb{Z}$ in the discrete case, and as $\mathbb{Q}$ and appropriate operations and relations over $\mathbb{Q}$ in the dense case. We call these structures *MTL-Π-structures*.

[2] The definition of Π given covers only the fragment of bounded universal Horn formulae but it can be extended to full first-order temporal logic.

The resulting formulae contain explicit quantifiers for time variables only such that it suffices to inspect term generated structures with respect to $\Pi(\Sigma)$ as models of the translated temporal logic programs, this is, $\mathscr{A}$-structures with respect to the algebras

$$\mathscr{A}_1 = (\mathbb{Z}, \mathscr{T}_\Sigma, 0, +1, -1, +, (f)_{f\in\Sigma}, =, \leqslant, <, =_{HB}),$$

respectively,

$$\mathscr{A}_2 = (\mathbb{Q}, \mathscr{T}_\Sigma, 0, +1, -1, +, (f)_{f\in\Sigma}, =, \leqslant, <, =_{HB}),$$

where $(\mathscr{T}_\Sigma, (f)_{f\in\Sigma}, =_{HB})$ is the ground term Σ-algebra. We identify $\mathscr{A}$-structures with subsets of the $\mathscr{A}$-base, which is defined as the set of all $\mathscr{A}$-instances of atomic Σ-formulae.

Proposition 4.1. *Let X_1 be a set of discrete and X_2 be a set of dense bounded universal Horn formulae and A_1 a discrete and A_2 a dense bounded universal goal. The following are equivalent*:

1. $X_i \models A_i$
2. $\Pi(X_i) \models_\Pi \Pi(A_i)$
3. $\Pi(X_i) \models_{\mathscr{A}_i} \Pi(A_i)$,

where $\models_\Pi$ denotes the validity in all MTL-Π-structures (of appropriate time domain) and $\models_{\mathscr{A}_i}$ the validity in all $\mathscr{A}_i$-structures.

Proof. (1) $\Leftrightarrow$ (2) can be shown in lines of [21, 20].

(2) $\Rightarrow$ (3) is obvious. (3) $\Rightarrow$ (2) can be seen as follows: Each MTL-Π-structure interprets, by definition, the sort *time* and operations and relations over *time* in the same, predefined way. These symbols are disjoint to those originating from the temporal signature, i.e. to symbols $p : time s_1 \ldots s_n \in \Pi(P)$ if $p : s_1 \ldots s_n \in P$ (respectively, $f \in F$). Since bounded universal Horn formulae contain explicit quantifiers over the integers (respectively, rationals) only, each satisfiable existential subformulae is satisfied by an element of $\mathbb{Z}$ (respectively, $\mathbb{Q}$). The implicit quantifiers are universal and range over sorts from S, which are disjoint to *time* such that each formula satisfied by an $d \in \mathscr{D}_s, s \in S$, can be also satisfied by a reachable $d' = eval(t)$ for some $t \in \mathscr{T}_{\Sigma_s}$. These observations allow already to construct for each MTL-Π-structure $\mathscr{M}$ being a model of a set of bounded universal Horn formulae X a model $\mathscr{M}'$ with $\mathscr{T}_\Sigma$ as domain for sorts of S and the usual term operations as operations from F, i.e. an $\mathscr{A}_i$-model, in lines of the construction for universal theories of classical logic. $\square$

The translation of bounded universal Horn formulae results into implication formulae with constraints

$$D ::= A \mid \forall x(C \rightarrow D) \mid D \leftarrow G$$

and goals

$$G ::= C \mid A \mid G \wedge G \mid \exists x G \mid \forall y(\{c^- \leqslant^- y \leqslant^+ c^+\} \rightarrow G) \mid$$
$$\exists x(G \wedge \forall y(\{t_1 \leqslant^- y \leqslant^+ t_2\} \rightarrow G)),$$

where A ranges over atoms, C over sets of constraints over $\mathscr{A}_i$, $c^-, c^+ \in \mathbb{Z}$ (respectively, $c^-, c^+ \in \mathbb{Q}$), x, y being variables ranging over $\mathbb{Z}$ (respectively, $\mathbb{Q}$) and $t_i = x + b$ or $t_i = b$ for some $b \in \mathbb{Z}$ (respectively, $b \in \mathbb{Q}$).[3] Since $((B \rightarrow A) \leftarrow C) \leftrightarrow (A \leftarrow B \wedge C)$ and $((A \leftarrow B) \leftarrow C) \leftrightarrow (A \leftarrow (B \wedge C))$, we restrict our attention to constraint implication formulae of the form $A \leftarrow C \wedge G$. We call them *extended (constraint logic programming) Horn formulae* or short *extended CLP-formulae* (respectively, *extended (constraint logic programming) goals* or *extended CLP-goals*).

5. Constraint logic interpretation

Constraint logic programs have been proposed in [59] in order to enhance the expressiveness of logic programs. They generalize Horn logic programs over term algebras to Horn programs over arbitrary structures being solution compact and which have satisfaction complete theories. A Σ-algebra $\mathscr{A}$ is called *solution compact* if each element d in $\mathscr{A}$ can be defined by a (possibly infinite) conjunction of constraints, and the complement of each constraint C can be defined by a (possibly infinite) disjunction of constraints [61]. Given an algebra $\mathscr{A}$ and a theory $\mathscr{T}$, they say to *correspond* if

1. $\mathscr{A}$ is a model of $\mathscr{T}$, and
2. for every constraint C, $\mathscr{A} \models \tilde{\exists} C$ iff $\mathscr{T} \models \tilde{\exists} C$.

$\mathscr{T}$ is *satisfaction complete* with respect to $\mathscr{A}$ if for every constraint C, either $\mathscr{T} \models \tilde{\exists} C$ or $\mathscr{T} \models \neg\tilde{\exists} C$. The algebras $\mathscr{A}_i$ are solution compact and have satisfaction complete theories. Solution compactness is easy to see, a satisfaction complete theory for $\mathscr{A}_1$ can be obtained by combination of Clark's axiomatization of unification [31] and a complete axiomatization of Presburger arithmetics. For $\mathscr{A}_2$ satisfaction completeness follows from completeness of the theory of real-closed fields and from Clark's axiomatization of unification [31], which defines a satisfaction complete theory for the ground term algebra $\mathscr{T}_\Sigma$.

Almost all of the classical results for logic programs have been generalized to the constraint logic case. The notion of Herbrand-base, Herbrand-model, least and greatest Herbrand-model of a program P is replaced by that of $\mathscr{A}(\Sigma)$-base, $\mathscr{A}(\Sigma)$-model, least and greatest $\mathscr{A}(\Sigma)$-model of a program P over a given structure $\mathscr{A}$. The functional semantics defined by fixed points of a function T_P mapping from and into the Herbrand base of P is generalized and is given by fixed points of a function $T_{(P,\mathscr{A})}$ mapping from and into the $\mathscr{A}(\Sigma)$-base of P. The operational semantics, this is, the derivation of goals

[3] According to the translation given above t_i are either of the form $t_i = x$ or $t_i = b$, $b \in \mathbb{Z}$ (respectively, $b \in \mathbb{Q}$). In Section 8, however, a slightly modified translation for discrete time structures is used, which introduces also terms of the form $x + b$ for the t_i.

from programs, is substituted by $(P,\mathscr{A})$-derivations, which generalize SLD-derivations. The condition for a resolution step

$$A_1,\dots,A_n \vdash_{SLD} (A_1,\dots,A_{i-1},B_1,\dots,B_m,A_{i+1},\dots,A_n)\theta$$

is the existence of a formula $A \leftarrow B_1,\dots,B_m$ in the program P such that A_i and A unify with the most general unifier θ, which is replaced by a similar condition for a $(P,\mathscr{A})$-derivation step

$$C,A_1,\dots,A_n \vdash_{(P,\mathscr{A})} \tilde{C},A_1,\dots,A_{i-1},B_1,\dots,B_m,A_{i+1},\dots,A_n,$$

namely that there is a formula $A \leftarrow C',B_1,\dots,B_n$ in the program P and $\tilde{C} = \{A_i = A\} \cup C \cup C'$ being $\mathscr{A}$-satisfiable, where C, C' are constraints over $\mathscr{A}$. Given a set of constraints C (over an algebra $\mathscr{A}$), $[\![C]\!]$ denotes the set of its solutions, i.e. $[\![C]\!] = \{\alpha : \mathscr{V} \to \mathscr{A} \mid \mathscr{A} \models C\alpha\}$, and $[\![C]\!]_V$ for a set of variables V, the solutions of C restricted to V, i.e. $[\![C]\!]_V = \{\alpha|_V \mid \alpha \in [\![C]\!]\}$, where $\alpha|_V(x) = \alpha(x)$ if $x \in V$ and $\alpha|_V(x) = x$ otherwise. C is called *satisfiable* if $[\![C]\!] \neq \emptyset$.

As the first step towards semantical characterizations of extended CLP-formulae we generalize the fixed point operator $T_{(P,\mathscr{A}_i)}$ in order to handle bounded $\forall$ quantifiers in their bodies:

$$\begin{aligned} T_{(P,\mathscr{A}_i)}(S) = \{d \in \mathscr{A}_i - \text{base} \mid & \text{there is a formula } A \leftarrow C \wedge G \text{ in } P, \text{ an} \\ & \mathscr{A}_i\text{-assignment } \alpha \text{ such that } \mathscr{A}_i \models d\alpha = A\alpha, \\ & \mathscr{A}_i \models c\alpha, \text{ and } S \models G\alpha\}, \end{aligned}$$

where $S \models G$ is defined for closed[4] extended goal formulae by

1. $S \models A$ iff $A \in S$, for a ground atomic formula A,
2. $S \models A \wedge B$ iff $S \models A$ and $S \models B$,
3. $S \models \forall y(\{c^- \preccurlyeq^- y \preccurlyeq^+ c^+\} \to G)$ iff for all $\mathscr{A}_i$-assignments α with $c^- \preccurlyeq^- \alpha(y) \preccurlyeq^+ c^+$: $S \models G\alpha$,
4. $S \models \exists x G$ iff there is an $\mathscr{A}_i$-assignment α to x such that $S \models G\alpha$.

$T_{(P,\mathscr{A}_i)}$ are well defined since subformulae $\forall y(\{x + c^- \preccurlyeq^- y \preccurlyeq^+ c^+\} \to G)$ occur in extended CLP-formulae always in the scope of an $\exists x$ quantifier.

Lemma 5.1. *$T_{(P,\mathscr{A}_i)}$ is monotonic, $i = 1,2$.*

Following [105, 9], we characterize the semantics in terms of canonical models and fixed points of the $T_{(P,\mathscr{A}_i)}$-operator.

Lemma 5.2 ($\mathscr{A}$-Model-lemma). *Let P_1 be a discrete extended CLP-program and P_2 a dense extended CLP-program.*

1. *I is an $\mathscr{A}_i$-model of P_i iff $T_{(P_i,\mathscr{A}_i)}(I) \subseteq I$.*
2. *There exists a least $\mathscr{A}_i$-model of P_i, $lm(P_i,\mathscr{A}_i)$, which is equal to $lfp(T_{(P_i,\mathscr{A}_i)})$.*

[4] A formula A is called *closed* if all variables occurring in A are bounded by quantifiers.

3. *There exists a greatest $\mathscr{A}_i$-model of P_i, $gm(P_i, \mathscr{A}_i)$, which is equal to $gfp(T_{(P_i,\mathscr{A}_i)})$,*

where $lfp(T_{(P_i,\mathscr{A}_i)})$ denotes the least fixed point and $gfp(T_{(P_i,\mathscr{A}_i)})$ the greatest fixed point of $T_{(P_i,\mathscr{A}_i)}$.

Proof. 1. Follows directly from the definition of $T_{(P_i,\mathscr{A}_i)}$ and the consequence relation $\models$ in $\mathscr{A}_i$-models.

2 and 3. From the monotonicity of $T_{(P_i,\mathscr{A}_i)}$ we conclude that $T_{(P_i,\mathscr{A}_i)}$ has a least fixed point $lfp(T_{(P_i,\mathscr{A}_i)})$ and a greatest fixed point $gfp(T_{(P_i,\mathscr{A}_i)})$. Thereby 2 and 3 follow directly from 1. □

The characterizations above can be lifted to the temporal level associating to each MTL-$\Pi(\Sigma)$-structure $\Pi(\mathscr{M})$ a Σ-structure $\mathscr{M}$ called its *corresponding structure* in a bijective way: This structure has the same domain as $\Pi(\mathscr{M})$, $\mathscr{D}_s^{\Pi(\mathscr{M})} = \mathscr{D}_s^{\mathscr{M}}$ for all $s \in S$, the same functions, $f^{\Pi(\mathscr{M})} = f^{\mathscr{M}}$ for all $f \in F$, and the predicates defined by $p^{\Pi(\mathscr{M})}(t, d_1, \ldots, d_n)$ is true iff $p_t^{\mathscr{M}}(d_1, \ldots, d_n)$ is true for predicate symbols $p : s_1, \ldots, s_n$ in Σ, $d_i \in \mathscr{D}_{s_i}$, and time points t.

Formally, this correspondence defines a functor Π from the category of MTL-structures over Σ consisting of the MTL-structures as objects and appropriate morphisms (cf. [103, 104]) to the category of MTL-Π-structures with MTL-Π-structures over $\Pi(\Sigma)$ as objects and suitable morphisms.

Corollary 5.3. *Let P be a set of bounded universal Horn formulae.*
1. *There exists a least MTL-model of P, namely*

$$lm(P) = \Pi^{-1}(lm(\Pi(P), \mathscr{A}_i)) = lfp(T_P) \text{ and}$$

2. *There exists a greatest MTL-model of P, namely*

$$gm(P) = \Pi^{-1}(gm(\Pi(P), \mathscr{A}_i)) = gfp(T_P),$$

where T_P is defined by $T_P = \Pi^{-1} \circ T_{(\Pi(P),\mathscr{A}_i)} \circ \Pi$, Π maps a MTL-structure into its corresponding $\mathscr{A}_i$-structure, Π^{-1} denotes the inverse of Π, and $\Pi(P)$ denotes the translation of the program P.

Proof. The proof follows essentially the lines of [20, 21] defining a fixed point operator T_P on the level of temporal (Herbrand) structures, showing $T_P = \Pi^{-1} \circ T_{(P,\mathscr{A}_i)} \circ \Pi$ and lifting the least model and fixed point results to the temporal level. For the greatest fixed point characterization, we need to define the completion P^* of a bounded universal program P both on the temporal and classical logic level. Using $T_P = \Pi^{-1} \circ T_{(P,\mathscr{A}_i)} \circ \Pi$, we can then show that the $\mathscr{A}$-model-lemma implies the existence of a greatest model with respect to the completion P^* on the classical and temporal logic level. □

6. Proving bounded universal goals

Standard proof methods in temporal logics rely on recursive characterizations of $\Box$, $\Diamond$, $\mathcal{S}$ and $\mathcal{U}$ operators, e.g., those presented in [107, 3, 8]:

$$\begin{aligned}
\Box A &\leftrightarrow A \wedge \circ\Box A \\
\Diamond A &\leftrightarrow A \vee \circ\Diamond A \\
A\mathcal{S}B &\leftrightarrow \bullet(B \vee (A \wedge A\mathcal{S}B)) \\
A\,\mathcal{U}\,B &\leftrightarrow \circ(B \vee (A \wedge A\,\mathcal{U}\,B))
\end{aligned}$$

They are usually combined with loop checking mechanisms in the propositional case or, in the first-order case, with some kind of induction rules. A naive attempt to exploit this technique would unfold goals of the form

$$\Box_c A \quad \text{into} \quad \bigwedge_{i=1}^{c} \circ^i A$$

ignoring the structure of the formulae to be proven and leads to very inefficient derivations. Besides this, it is not obvious how the proof methods based on the unfolding approach can be adapted for dense time structures.

Example 6.1. Consider the translation of an excerpt of the slightly simplified (discrete) bounded universal Horn program listed in Fig. 1.

$$\begin{array}{ll}
\Box_{[-2000,-1600]} & \text{salesman(john)} \\
\Box_{[-1599,-1100]} & \text{manager_of(john, sales)} \\
\Box_{[-1099,-600]} & \text{manager_of(john, development)} \\
\Box_{[-599,-1]} & \text{manager_of(john, board)} \\
\multicolumn{2}{l}{\Box(\text{manager(P)} \leftarrow \text{manager_of(P, D)})} \\
\multicolumn{2}{l}{\Box(\text{salary(P, S)} \leftarrow \text{manager_of(P, D)} \wedge \text{S} = f(\text{D}))}
\end{array}$$

The function f computes the salary for every manager of a given department which is in general a rather complex operation but one which can be computed in an uniform way for large intervals. [5] Queries for salary predicate (respectively, for the manager predicate) can therefore be proven uniformly for large intervals. But proving

$$\leftarrow \Box_{[-1500,-1]}(\text{manager(john)} \wedge \exists S(\text{salary(john},S) \wedge S \geqslant 100000))$$

[5] The factors influencing the salary of a manager are assumed not to change very often.

via the unfolding approach means proving

$$\leftarrow \bigwedge_{t=-1500}^{-1} \circ^t(\text{manager(john)} \wedge \exists S(\text{salary(john}, S) \wedge S \geqslant 100000))$$

without be able to use the uniform proofs for the manager and the salary predicate.

Our intention is to explore these uniform proofs and present an operational semantics for bounded universal Horn formulae which efficiency is comparable to SLD-derivations, i.e., each of the basic steps can be performed in linear time and which admits an intuitive procedural interpretation.

7. Simple (metric) temporal logic programs

As a first step towards such an operationalization, we consider a fragment that can be handled within the constraint logic programming framework [25]. The goals of this fragment are called *simple MTL-goals* and are defined by

$$G ::= \varepsilon \mid A \mid \Diamond_I G \mid G \wedge G,$$

the Horn formulae called *simple MTL-Horn formulae* by

$$D ::= A \mid \Box_I D \mid D \leftarrow G,$$

where I denotes an interval, ε the empty goal and A ranges over atoms.

In the discrete case this class can be also defined as follows: Simple MTL-goals by

$$G ::= \varepsilon \mid A \mid \circ G \mid \bullet G \mid \Diamond_c G \mid G \wedge G$$

and *simple MTL-Horn formulae* by

$$D ::= A \mid \circ D \mid \bullet D \mid \Box_c D \mid D \leftarrow G.$$[6]

In the rest of this section we focus on discrete MTL-programs but use also the first of the equivalent definitions above whenever it helps to simplify the presentation. As we will see later on, the results of the section can be easily adapted for the dense case as well.

The first (obvious) observation (of the first definition) is that sequences of universal $\Box_{I_1} \cdots \Box_{I_n}$ and existential operators $\Diamond_{I_1} \cdots \Diamond_{I_m}$ can be normalized to $\Box_{\Sigma_{i=1}^n I_i}$ (respectively, $\Diamond_{\Sigma_{i=1}^m I_i}$) and $\Box_{[0,0]} A \leftrightarrow A$ such that it is sufficient to consider simple MTL-Horn formulae of the form

$$\Box_{[d_1^-, d_1^+]} (\Box_{[d_2^-, d_2^+]} (\ldots \Box_{[d_{n+1}^-, d_{n+1}^+]} A \leftarrow B_n \ldots B_2) \leftarrow B_1). \tag{1}$$

[6] The programs in Examples 3.1 and 3.2 are simple MTL-programs.

$$\pi(A', \textstyle\sum_{i=1}^{m+1} x_i + b) \leftarrow \bigcup_{i=1}^{m+1}\{c_i^- \leqslant x_i \leqslant c_i^+\} \cup C' \wedge \bigwedge_{i=1}^{m} \pi(B_i', \sum_{j=1}^{i} x_j)$$
$$\underline{\leftarrow C \wedge \pi(A, \bar{y} + c) \wedge \textstyle\bigwedge_{i=0}^{n} \pi(B_i, \bar{y}_{n-i})}$$
$$\leftarrow (\{c_1^- \leqslant \bar{y} + \textstyle\sum_{i=m+1}^{2} - x_i + c - b \leqslant c_1^+\} \cup \bigcup_{i=2}^{m+1}\{-c_i^+ \leqslant -x_i \leqslant -c_i^-\} \cup$$
$$C \cup C' \wedge \textstyle\bigwedge_{i=1}^{m} \pi(B_i', \bar{y} + \sum_{j=m+1}^{i+1}(-x_j) + c - b) \wedge \bigwedge_{i=0}^{n} \pi(B_i, \bar{y}_{n-i}))\theta,$$

where θ is the mgu of A and A', $\bar{y}_{n-i}$ denotes $\sum_{j=1}^{n-i} y_j$, and $\bar{y} = \bar{y}_n$.

Fig. 4. MTL-resolution for simple MTL-goals.

Their translation results into formulae of the form

$$\pi\left(A', \sum_{i=1}^{m+1} x_i + b\right) \leftarrow \bigcup_{i=1}^{m+1} \{c_i^- \leqslant x_i \leqslant c_i^+\} \wedge \bigwedge_{i=1}^{m} \pi\left(B_i', \sum_{j=1}^{i} x_j\right) \tag{2}$$

with $b \in \mathbb{Z}$, $c_i^-, c_i^+ \in \mathbb{Z} \cup \{-\infty, \infty\}$, and x_i possibly constrained to be 0, i.e., $c_i^- = c_i^+ = 0$. Moreover, translation of goals yields $(\Pi(P), \mathscr{A}_1)$-goals that have the form

$$\leftarrow C \wedge \pi(A, \bar{y} + c) \wedge \bigwedge_{i=0}^{n} \pi(B_i, \bar{y}_{n-i}), \tag{3}$$

with $c \in \mathbb{Z}$, $\bar{y} = y_1 + \cdots + y_n$, $\bar{y}_i = y_1 + \cdots + y_i$, A being an atom, C a set of inequalities and B_i goals, $i = 0, \ldots, n$. These normal forms allow to simplify SLD-resolution with constraints to the SLD-resolution variant listed in Fig. 4.

Theorem 7.1 (Soundness, completeness). *Let P be a simple MTL-program and G a simple MTL-goal. Then*

(Completeness) *If* $\leftarrow G \vdash_{(\Pi(P), \mathscr{A}_1)} \leftarrow G'$, *then* $\leftarrow G \vdash_{MTL} \leftarrow G'$,

(Soundness) *If* $\leftarrow G \vdash_{MTL} \leftarrow G'$, *then* $\leftarrow G \vdash_{(\Pi(P), \mathscr{A}_1)} \leftarrow G'$, *where* $\vdash_{MTL}$ *stands for derivable using the MTL-resolution rule.*

Proof. Notice that the substitution for x_1 in the MTL-rule keeps the form of the translated goal as described in (3). Therefore, it is sufficient to prove that the MTL-resolution rule is equivalent to the corresponding $(\Pi(P), \mathscr{A}_1)$-derivation step, which yields

$$\leftarrow \left(C \cup C' \cup \bigcup_{i=1}^{m+1} \{c_i^- \leqslant x_i \leqslant c_i^+\} \cup \underbrace{\left\{ \sum_{i=1}^{m+1} x_i + b = \bar{y} + c \right\}}_{e} \right.$$
$$\left. \wedge \bigwedge_{i=1}^{m} \pi\left(B_i', \sum_{j=1}^{i} x_j\right) \wedge \bigwedge_{i=0}^{n} \pi(B_i, \bar{y}_{n-i}) \right)\theta.$$

By rewriting the equation e with x_1 as subject we get the equation $x_1 = \bar{y} - \sum_{i=2}^{m+1} x_i + c - b$. Since x_1 doesn't occur in C, C', $\bigcup_{i=2}^{m+1}\{c_i^- \leqslant x_i \leqslant c_i^+\}$, and $\bigwedge_{i=0}^{n} \pi(B_i, \bar{y}_{n-i})$, we get by elimination of x_1 the goal

$$\leftarrow \left(C \cup C' \cup \left\{ c_1^- \leqslant \bar{y} + \sum_{i=2}^{m+1} - x_i + c - b \leqslant c_1^+ \right\} \cup \bigcup_{i=2}^{m+1} \{c_i^- \leqslant x_i \leqslant c_i^+\} \right.$$

$$\left. \wedge \bigwedge_{i=1}^{m} \pi \left(B_i', \underbrace{\sum_{j=2}^{i} (x_j) + \bar{y} - \sum_{j=2}^{m+1} x_j + c - b}_{\bar{y} + \sum_{j=m+1}^{i+1} - x_j + c - b} \right) \wedge \bigwedge_{i=0}^{n} \pi(B_i, \bar{y}_{n-i}) \right) \theta,$$

which is equivalent to the conclusion of the MTL-resolution rule. □

As Corollary of 7.1 and the completeness of $(\Pi(P), \mathscr{A}_1)$-derivations we obtain

Corollary 7.2 (Brzoska [23]). *MTL-resolution in conjunction with constraint checking over $\mathscr{A}_1$ is sound and complete for proving simple MTL-goals from simple MTL-programs.*

MTL-resolution is not sufficient for proving goals containing $\Box_c$, $\mathscr{S}$, and $\mathscr{U}$ operators.

Example 7.3. Consider the simplified MTL-program from Fig. 1 with the translation

salesman(x, john) $\leftarrow \{-20 \leqslant x \leqslant -16\}$

manager_of(x, john, sales) $\leftarrow \{-15 \leqslant x \leqslant -11\}$

manager_of(x, john, development) $\leftarrow \{-10 \leqslant x \leqslant -6\}$

manager_of(x, john, board) $\leftarrow \{-5 \leqslant x \leqslant -1\}$

manager(x, Person) $\leftarrow$ manager_of(x, Person, Department)

Using the MTL-resolution mechanism (or the CLP-derivation mechanism) we can prove the goals [7]

$\leftarrow \Box_{[-15,-11]}$manager(john) $\equiv_\Pi \leftarrow \forall x(\{-15 \leqslant x \leqslant -11\} \rightarrow$ manager(x, john))

$\leftarrow \Box_{[-10,-6]}$manager(john) $\equiv_\Pi \leftarrow \forall x(\{-10 \leqslant x \leqslant -6\} \rightarrow$ manager(x, john))

$\leftarrow \Box_{[-5,-1]}$manager(john) $\equiv_\Pi \leftarrow \forall x(\{-5 \leqslant x \leqslant -1\} \rightarrow$ manager(x, john))

[7] $A \equiv_\Pi B$ denotes $\Pi(A) = B$.

but not [8]

$$\leftarrow \Box_{[-15,-6]}\text{manager}(\text{ john}) \equiv_{\Pi} \leftarrow \forall x(\{-15 \leqslant x \leqslant -6\} \rightarrow \text{manager}(x, \text{john}))$$

$$\leftarrow \Box_{[-15,-1]}\text{manager}(\text{ john}) \equiv_{\Pi} \leftarrow \forall x(\{-15 \leqslant x \leqslant -1\} \rightarrow \text{manager}(x, \text{john}))$$

$$\leftarrow \Box_{[-10,-1]}\text{manager}(\text{ john}) \equiv_{\Pi} \leftarrow \forall x(\{-10 \leqslant x \leqslant -1\} \rightarrow \text{manager}(x, \text{john}))$$

$$\begin{aligned}\leftarrow \text{manager}(\text{john}) \ \mathcal{S} \ \text{salesman}(\text{john}) \equiv_{\Pi} \leftarrow \exists x(\{x \leqslant -1\} \wedge \text{salesman}(x, \text{john}) \\ \wedge \forall y(\{x+1 \leqslant y \leqslant -1\} \rightarrow \text{manager}(y, \text{john}))),\end{aligned}$$

which also follow from the program. They are direct consequences of the first three goals and the clause $\Box_{[-20,-16]}$ salesman(john) but are not provable by a single MTL-derivation.

7.1. Solving systems of linear inequalities over $\mathscr{A}_i$

In 1824 Fourier proposed the first algorithm for solving linear arithmetic constraints. Apart from historical interest – its complexity is exponential in general – it has interesting theoretical properties [74]. We use the algorithm to eliminate existential quantifiers and to prove the LR-lemma allowing to eliminate bounded universal quantifiers. This lemma is the key of our proof method for bounded universal goals. We will show that the algorithm has linear time complexity for constraint systems generated during MTL-derivations. It is thereby the most promising method for satisfiability checking during derivations of temporal goals. Our presentation follows essentially the lines of [74], but we extend their Fourier's algorithm to strict inequalities, $<$, needed in the dense time part of the paper.

Let P be a set of inequalities and x a variable in P. Let, for convenience, arrange P by equivalent algebraic manipulation into the form

$$\begin{array}{ll} l_i \leqslant x & i=1,\dots,p \\ l'_i < x & i=1,\dots,p' \\ x \leqslant r_j & j=1,\dots,q \\ x < r'_j & j=1,\dots,q' \\ d_l \leqslant_l 0 & l=1,\dots,s, \end{array} \tag{4}$$

where $p, p', q, q', s \geqslant 0$ and $l_i, l'_i, r_j, r'_j, d_l$ does not contain x, $\leqslant_l \in \{<, \leqslant\}$.

A *Fourier step eliminating x from P* transforms P into

$$\begin{array}{ll} l_i \leqslant r_j & i=1,\dots,p;\ j=1,\dots,q \\ l_i < r'_j & i=1,\dots,p;\ j=1,\dots,q' \\ l'_i < r_j & i=1,\dots,p';\ j=1,\dots,q \\ l'_i < r'_j & i=1,\dots,p';\ j=1,\dots,q' \\ d_l \leqslant_l 0 & l=1,\dots,s. \end{array}$$

[8] The translation of the $\mathcal{S}$ operator used below is sound for discrete time only.

Fourier's algorithm consists of the following steps: select a variable, say x, to eliminate, arrange the inequalities into a form appropriate for elimination of x, apply a Fourier elimination step eliminating x. The algorithm terminates if one of the elimination steps generate a contradictory inequality $0 \preccurlyeq c$ or if all variables have been eliminated. In the first case, the original inequalities were unsatisfiable, and in the second case we infer their satisfiability. In order to simplify notation, we use $\preccurlyeq_1 \downarrow \preccurlyeq_2$ to denote $\leqslant$ if $\preccurlyeq_1 = \preccurlyeq_2 = \leqslant$, and $<$ otherwise.

Lemma 7.4. *Let P be a set of inequalities and let P' be derived from P by a Fourier step eliminating the variable x from P, then $\exists x P \leftrightarrow P'$.*

Proof. Let P be w.l.o.g. of the form given in (4) and let α denote its solution. Then

$$\bar{\alpha}(l_i) \leqslant \alpha(x) \leqslant \bar{\alpha}(r_j), \quad i=1,\dots,p; \; j=1,\dots,q \tag{5}$$

$$\bar{\alpha}(l_i) \leqslant \alpha(x) < \bar{\alpha}(r_j'), \quad i=1\dots,p; \; j=1,\dots,q' \tag{6}$$

$$\bar{\alpha}(l_i') < \alpha(x) \leqslant \bar{\alpha}(r_j), \quad i=1,\dots,p; \; j=1,\dots,q' \tag{7}$$

$$\bar{\alpha}(l_i') < \alpha(x) < \bar{\alpha}(r_j'), \quad i=1,\dots,p; \; j=1,\dots,q' \tag{8}$$

$$\bar{\alpha}(d_i) \preccurlyeq_i 0, \quad i=1,\dots,s. \tag{9}$$

"$\rightarrow$": Assume $\exists x P$. Then there exists an α satisfying (4) and, as a consequence, (5)–(9) and also P'. Since x does not occur in P', the choice for x does not influence the satisfiability of P', and consequently $\exists x P \rightarrow P'$.

"$\leftarrow$": Assume P' is satisfiable with α. Then

$$\bar{\alpha}(l_i) \leqslant \bar{\alpha}(r_j), \quad i=1,\dots,p; \; j=1,\dots,q$$

$$\bar{\alpha}(l_i) < \bar{\alpha}(r_j'), \quad i=1,\dots,p; \; j=1,\dots,q'$$

$$\bar{\alpha}(l_i') < \bar{\alpha}(r_j), \quad i=1,\dots,p'; \; j=1,\dots,q$$

$$\bar{\alpha}(l_i') < \bar{\alpha}(r_j'), \quad i=1,\dots,p'; \; j=1,\dots,q'$$

and consequently

$$m_1 = \max_{i=1,\dots,p} \{\bar{\alpha}(l_i)\} \leqslant \min_{j=1,\dots,q} \{\bar{\alpha}(r_j)\} \tag{10}$$

$$\max_{i=1,\dots,p} \{\bar{\alpha}(l_i)\} < \min_{j=1,\dots,q'} \{\bar{\alpha}(r_j')\} \tag{11}$$

$$m_2 = \max_{i=1,\dots,p'} \{\bar{\alpha}(l_i')\} < \min_{j=1,\dots,q} \{\bar{\alpha}(r_j)\} \tag{12}$$

$$\max_{i=1,\dots,p'} \{\bar{\alpha}(l_i')\} < \min_{j=1,\dots,q'} \{\bar{\alpha}(r_j')\}. \tag{13}$$

We extend α for x by $\alpha(x) = m_1$, if $m_1 > m_2$, and for $m_1 \leqslant m_2$ by

$$\alpha(x) = m = \tfrac{1}{2}(m_2 + \min_{l=1,\dots,q;\, j=1,\dots,q'} \{\bar{\alpha}(r_l), \bar{\alpha}(r_j')\}).$$

MTL-resolution generates only a restricted class of constraints to be checked for satisfiability which can be associated with trees. A set T of terms of the form $\sum x_i + c$

with pairwise distinct variables x_i and $c \in \mathbb{Z}$ [9] (respectively, $\mathbb{Q}$) *defines a tree* if for every variable x in $Var(T)$ there is an unique prefix $\sum_{i=1}^{n} x_i$ in T with $x_n = x$ (*unique prefix property*). [10] It is easy to see that if T defines a tree, then $(Var(T), \leqslant)$ is a tree, where $\leqslant$ is defined by: $x \leqslant y$ iff there exist $\bar{x}, \bar{y} \in T$ such that $\bar{x} + x$ and $\bar{y} = \bar{x} + x + y$.

A system of inequalities C is called an *extended tree constrain system*, short an ETCS, with respect to a tree $(Var(C), \leqslant)$ if it is of the form $C' \cup E$ with

$$C' \subseteq \{c^- \preccurlyeq_x^- x, x \preccurlyeq_x^+ c^+ \mid x \text{ is a variable}\} \cup \{c^- \preccurlyeq_{\bar{x}}^- \bar{x}, \bar{x} \preccurlyeq_{\bar{x}}^+ c^+ \mid \text{for paths } \bar{x} \text{ in } (Var(C), \leqslant) \text{ starting from the root}\}$$

and

$$E \subseteq \{y + c^- \preccurlyeq_{y,y'}^- y', y' \preccurlyeq_{y,y'}^+ y + c^+ \mid \text{for paths } \bar{x} + y, \bar{x} + y' \text{ in } (Var(C), \leqslant) \text{ starting from the root}\},$$

$\preccurlyeq \in \{<, \leqslant\}$, $c^-, c^+ \in \mathbb{Z}$ (respectively, $c^-, c^+ \in \mathbb{Q}$), and $\bar{x} = x_1 + \cdots + x_n$. An extended tree constrain system with an empty E, $E = \emptyset$, is also called *tree constraint system*, short a TCS. MTL-resolution generates only extended tree constraint systems.

Lemma 7.5. *If $\Pi(G) \vdash_{MTL} G'$, then $G'\theta_{G'}$ contains only extended tree constraint systems, where $\theta_{G'}(x) = -x$ if x occurs with negative sign in G' and $\theta_{G'}(x) = x$ otherwise.*

We use this observation for a specialized method for satisfiability checking during MTL-derivations based on Fourier's algorithm, which eliminates variables being leaves of the tree underlying the ETCS of interest. This strategy keeps the ETCS property invariant, and since the coefficients of the variables occurring in ETCSs are either 1 or 0, they can be eliminated without algebraic manipulations on the coefficients. Checking for satisfiability over the integers coincides thereby with satisfiability checking over the reals. A rule based formulation of the method is listed in Figs. 6 and 7 for systems containing strict inequalities. A simplified version for systems without strict inequalities, which are sufficient in the discrete case, is given in Fig. 5. We denote the simplification relation defined by rules (MLB), (MUB), (IUB), and (ILB) by $\rightarrow_{sc}$ and that by rules of $\rightarrow_{sc}$ and (E y i), $i = 1, 2$, by $\rightarrow_{esc}$. Their counterparts for systems with strict inequalities are denoted by $\rightarrow_{dsc}$ and $\rightarrow_{edsc}$. To ensure their termination we use an ordering $>$ on tuples $(c, \leqslant)$, $c \in \mathbb{Z}$ (respectively, $\mathbb{Q}$) defined by $(c_1, \preccurlyeq_1) < (c_2, \preccurlyeq_2)$ iff $c_1 < c_2$ or $c_1 = c_2$ and $\preccurlyeq_1 = <$, $\preccurlyeq_2 = \leqslant$.

[9] Terms $x_1 + \cdots + x_n + c$ are interpreted here as words over $\mathcal{V} \cup \mathbb{Z}$ with $+$ as concatenation on words.

[10] A similar notion called *prefix-stability* (respectively, *unique prefix property*) was introduced in the context of automated theorem proving in modal logics by translating into first-order logic [86] (respectively, [34]). This property of terms coding modal contexts of translated modal logic formulae guarantees that the unification under associativity of those terms always yields a finitely set of most general unifiers although unification under associativity is infinitary in general.

(ILB)	$C \cup \{c_1 \leqslant \bar{x} + y, y \leqslant c_3\} \rightarrow_{esc}$ $C \cup \{c_1 \leqslant \bar{x} + y, c_1 - c_3 \leqslant \bar{x}, y \leqslant c_3\}$ if for all $c_2 \leqslant \bar{x} \in C$, $c_1 - c_3 > c_2$.
(IUB)	$C \cup \{\bar{x} + y \leqslant c_1, c_3 \leqslant y\} \rightarrow_{esc}$ $C \cup \{\bar{x} + y \leqslant c_1, \bar{x} \leqslant c_1 - c_3, c_3 \leqslant y\}$ if for all $\bar{x} \leqslant c_2 \in C$, $c_2 > c_1 - c_3$.
(E y 1)	$C \cup \{c_1^- \leqslant \bar{x} + y \leqslant c_1^+, z + b \leqslant y \leqslant z' + b'\} \rightarrow_{esc} C \cup$ $\{c_1^- \leqslant \bar{x} + y \leqslant c_1^+, z + b \leqslant y \leqslant z' + b', c_1^- - b' \leqslant \bar{x} + z', \bar{x} + z \leqslant c_1^+ - b\}$ if for all $c_2^- \leqslant \bar{x} + z' \in C$, $c_1^- - b' > c_2^-$ or for all $\bar{x} + z \leqslant c_2^+ \in C$, $c_2^+ > c_1^+ - b$.
(E y 2)	$C \cup \{c_y^- \leqslant y \leqslant c_y^+, z + b \leqslant y \leqslant z' + b'\} \rightarrow_{esc} C \cup$ $\{c_y^- \leqslant y \leqslant c_y^+, z + b \leqslant y \leqslant z' + b', c_y^- - b' \leqslant z', z \leqslant c_y^+ - b\}$ if for all $c_{z'}^- \leqslant z' \in C$, $c_y^- - b' > c_{z'}^-$ or for all $z \leqslant c_z^+ \in C$, $c_z^+ > c_y^+ - b$.
(MLB)	$C \cup \{c_1 \leqslant \bar{x}, c_2 \leqslant \bar{x}\} \rightarrow_{esc} C \cup \{\max(c_1, c_2) \leqslant \bar{x}\}$
(MUB)	$C \cup \{\bar{x} \leqslant c_1, \bar{x} \leqslant c_2\} \rightarrow_{esc} C \cup \{\bar{x} \leqslant \min(c_1, c_2)\}$

Fig. 5. Satisfiability checking of extended tree constraint systems.

(ILB)	$C \cup \{c_1 \preccurlyeq_1 \bar{x} + y, y \preccurlyeq_3 c_3\} \rightarrow_{dsc}$ $C \cup \{c_1 \preccurlyeq_1 \bar{x} + y, c_1 - c_3 \preccurlyeq_1 \downarrow \preccurlyeq_3 \bar{x}, y \preccurlyeq_3 c_3\}$ if for all $c_2 \preccurlyeq_2 \bar{x} \in C$, $(c_1 - c_3, \preccurlyeq_1 \downarrow \preccurlyeq_3) > (c_2, \preccurlyeq_2)$.
(IUB)	$C \cup \{\bar{x} + y \preccurlyeq_1 c_1, c_3 \preccurlyeq_3 y\} \rightarrow_{dsc}$ $C \cup \{\bar{x} + y \preccurlyeq_1 c_1, \bar{x} \preccurlyeq_1 \downarrow \preccurlyeq_3 c_1 - c_3, c_3 \preccurlyeq_3 y\}$ if for all $\bar{x} \preccurlyeq_2 c_2 \in C$, $(c_2, \preccurlyeq_2) > (c_1 - c_3, \preccurlyeq_1 \downarrow \preccurlyeq_3)$.
(MLB1)	$C \cup \{c_1 \preccurlyeq_1 \bar{x}, c_2 \preccurlyeq_2 \bar{x}\} \rightarrow_{dsc} C \cup \{c_1 \preccurlyeq_1 \bar{x}\}$ if $c_1 > c_2$
(MLB2)	$C \cup \{c_1 \preccurlyeq_1 \bar{x}, c_1 \preccurlyeq_2 \bar{x}\} \rightarrow_{dsc} C \cup \{c_1 \preccurlyeq_1 \downarrow \preccurlyeq_2 \bar{x}\}$
(MUB1)	$C \cup \{\bar{x} \preccurlyeq_1 c_1, \bar{x} \preccurlyeq_2 c_2\} \rightarrow_{dsc} C \cup \{\bar{x} \preccurlyeq_1 c_1\}$, if $c_1 < c_2$
(MUB2)	$C \cup \{\bar{x} \preccurlyeq_1 c_1, \bar{x} \preccurlyeq_2 c_1\} \rightarrow_{dsc} C \cup \{\bar{x} \preccurlyeq_1 \downarrow \preccurlyeq_2 c_1\}$

Fig. 6. Satisfiability checking of tree constraint systems with strict inequalities.

Theorem 7.6 (Satisfiability checking). *Let C be an (extended) tree constraint system and let $\rightarrow$ denote $\rightarrow_{edsc}$ (respectively, $\rightarrow_{esc}$).*

(Termination) *There is no infinite chain $C = C_1 \rightarrow C_2 \rightarrow \ldots$*

(E y 1) $C \cup \{c_1^- \leqslant_1^- \bar{x}+y \leqslant_1^+ c_1^+, z+b \leqslant_y^- y \leqslant_y^+ z'+b'\} \rightarrow_{esc} C \cup$
$\{c_1^- \leqslant_1^- \bar{x}+y \leqslant_1^+ c_1^+, z+b \leqslant_y^- y \leqslant_y^+ z'+b',$
$c_1^- - b' \leqslant_1^- \downarrow \leqslant_y^+ \bar{x}+z', \bar{x}+z \leqslant_1^+ \downarrow \leqslant_y^- c_1^+ - b\}$
if for all $c_2^- \leqslant_2^- \bar{x}+z' \in C, (c_2^-, \leqslant_2^-) < (c_1^- - b', \leqslant_1^- \downarrow \leqslant_y^+)$,
respectively, for all $\bar{x}+z \leqslant_2^+ c_2^+ \in C$, $(c_2^+, \leqslant_2^+) > (c_1^+ - b, \leqslant_1^+ \downarrow \leqslant_y^-)$.

(E y 2) $C \cup \{c_y^- \leqslant_y^- y \leqslant_y^+ c_y^+, z+b \leqslant_1^- y \leqslant_1^+ z'+b'\} \rightarrow_{esc} C \cup$
$\{c_y^- \leqslant_y^- y \leqslant_y^+ c_y^+, z+b \leqslant_1^- y \leqslant_1^+ z'+b',$
$c_y^- - b' \leqslant_y^- \downarrow \leqslant_1^+ z', z \leqslant_1^- \downarrow \leqslant_y^+ c_y^+ - b\}$
if for all $c_{z'}^- \leqslant_{z'}^- z' \in C$, $(c_{z'}^-, \leqslant_{z'}^-) < (c_y^- - b', \leqslant_y^- \downarrow \leqslant_1^+)$,
respectively, for all $z \leqslant_z^+ c_z^+ \in C$, $(c_z^+, \leqslant_z^+) > (c_y^+ - b, \leqslant_1^- \downarrow \leqslant_y^+)$.

Fig. 7. Satisfiability checking for extended tree constraint systems with strict inequalities.

(Invariance) *If* $C \rightarrow C'$ *then* $[\![C]\!] = [\![C']\!]$, *where* $[\![C]\!]$ *denotes the solutions of* C, *i.e.* $[\![C]\!] = \{\alpha : \mathscr{V} \rightarrow \mathscr{A}_2 \mid C\alpha$ *is valid over* $\mathscr{A}_2\}$ *(respectively,* $[\![C]\!] = \{\alpha : \mathscr{V} \rightarrow \mathscr{A}_1 \mid C\alpha$ is valid over $\mathscr{A}_1\}$.

(Completeness) *If* C *is unsatisfiable, then there exists a* C' *such that* $C \xrightarrow{*} C'$ *and* C' *contains an inequality* $c_1 \leqslant_1 \bar{x} \leqslant_2 c_2$ *with* $c_1 > c_2$ *or* $c_1 = c_2$ *and one of the* $\leqslant_i$ *equals* $<$.

Proof. We show the theorem for $\rightarrow_{edsc}$ and obtain the corresponding results for $\rightarrow_{esc}$ by specialization.

(Termination) We define a well-founded ordering $>_{ec}$ on extended tree constraint systems such that the left-hand sides of the inference rules defining $\rightarrow_{edsc}$ are greater than the right-hand sides, respectively.

First we define the complexity $ec(C)$ of an ETCS by

$$ec(C) = c(C) \cup e_1(C) \cup e_2(C),$$

whereby

$$c(C) = \{d(\bar{x}+y) \mid \begin{array}{l} (1)\ \{c_1 \leqslant_1 \bar{x}+y, y \leqslant_3 c_3\} \subseteq C \text{ and for all} \\ c_2 \leqslant_2 \bar{x} \in C\text{: } (c_2, \leqslant_2) < (c_1 - c_3, \leqslant_1 \downarrow \leqslant_2), \text{ or,} \\ (2)\ \{\bar{x}+y \leqslant_1 c_1, c_3 \leqslant_3 y\} \subseteq C \text{ and for all} \\ \bar{x} \leqslant_2 c_2 \in C : (c_2, \leqslant_2) > (c_1 - c_3, \leqslant_1 \downarrow \leqslant_3)\} \end{array}$$

$$e_1(C) = \{d(\bar{x}+y), d(\bar{x}+y) \mid \begin{array}{l} (1)\ \{c_1^- \leqslant_1^- \bar{x}+y, y \leqslant_y^+ z'+b'\} \subseteq C \text{ and} \\ \text{for all } c_2^- \leqslant_2^- \bar{x}+z' \in C : (c_2^-, \leqslant_2) < \\ (c_1^- - b', \leqslant_1^- \downarrow \leqslant_y^+), \text{ or,} \\ (2)\ \{\bar{x}+y \leqslant_1^+ c_1^+, z+b \leqslant_y^- y\} \subseteq C \text{ and} \\ \text{for all } \bar{x}+z \leqslant_2^+ c_2^+ \in C : (c_2^+, \leqslant_2^+) > \\ (c_1^+ - b, \leqslant_y^- \downarrow \leqslant_1^+)\} \end{array}$$

$$e_2(C) = \{d(\bar{x}+y), d(\bar{x}+y) \mid \begin{array}{l}(1)\ \{c^- \preccurlyeq^- \bar{x} + y \preccurlyeq^+ c^+, c_y^- \preccurlyeq_y^- y, y \preccurlyeq_1^+ z' + b'\} \\ \subseteq C \text{ and for all } c_{z'}^- \preccurlyeq_{z'}^- z' \in C : (c_{z'}^-, \preccurlyeq_{z'}^-) > \\ (c_y^- - b', \preccurlyeq_y^- \downarrow \preccurlyeq_1^+), or, \\ (2)\ \{c^- \preccurlyeq^- \bar{x} + y \preccurlyeq^+ c^+, y \preccurlyeq_y^+ c_y^+, z + b \preccurlyeq_1^- y\} \subseteq C \\ \text{and for all } z \preccurlyeq_z^+ c_z^+ \in C : (c_z^+, \preccurlyeq_z^+) > \\ (c_y^+ - b, \preccurlyeq_1^- \downarrow \preccurlyeq_y^+)\}\end{array}$$

and $d(x_1 + \cdots + x_n) = n$. $>_{ec}$ is defined by

$$C_1 >_{ec} C_2 \quad \text{iff } |C_1| > |C_2| \text{ or}$$
$$|C_1| = |C_2| \text{ and } ec(C_1) \gg ec(C_2).$$

$|C|$ denotes the cardinality of C and $\gg$ the extension of $>$ on natural numbers to a multi-set ordering. $>_{ec}$ is well-founded, since it is defined by a lexicographical combination of two well-founded orderings [38].

Applications of the merge rules $(MLBi)$ and $(MUBi)$ decrease the cardinality of C. (ILB) and (IUB), however, do not change the cardinality of C and the sets $e_i(C)$. If $C \rightarrow_{edsc} C'$ by application of (ILB), then

$$\{c_1 \preccurlyeq_1 \bar{x} + y, y \preccurlyeq_3 c_3\} \subseteq C,$$
$$\{c_1 \preccurlyeq_1 \bar{x} + y, c_1 - c_3 \preccurlyeq_1 \downarrow \preccurlyeq_3 \bar{x}, y \preccurlyeq_3 c_3\} \subseteq C',$$

and for all $c_2 \preccurlyeq_2 \bar{x} \in C$, $(c_2, \preccurlyeq_2) < (c_1 - c_3, \preccurlyeq_1 \downarrow \preccurlyeq_3)$. Consequently, $c(C) = X \cup \{d(\bar{x} + y)\}$ and $c(C') = X \cup \{d(\bar{x})\}$ or $c(C') = X$. Since $d(\bar{x}) < d(\bar{x} + y)$, we have $c(C) \gg c(C')$. Similarly, if $C \rightarrow_{edsc} C'$ by application of (IUB), then $|C| = |C'|$ and $c(C) \gg c(C')$.

If $C_1 \rightarrow_{edsc} C_2$ by some of the (Eyi) rules, then one of the inequalities $c_1^- - b' \preccurlyeq_1^- \downarrow \preccurlyeq_y^+ \bar{x} + z'$, $\bar{x} + z \preccurlyeq_1^+ \downarrow \preccurlyeq_y^- c_1^+ - b$, $c_y^- - b' \preccurlyeq_y^- \downarrow \preccurlyeq_1^+ z'$ or $z \preccurlyeq_1^- \downarrow \preccurlyeq_y^+ c_y^+ - b$ are added to C_2. We have $ec(C_2) = c(C_2) \cup e_1(C_2) \cup e_2(C_2)$ with

$$c(C_2) \subseteq c(C_1) \cup \{d(\bar{x} + y)\} \text{ and}$$
$$e_i(C_2) = e_i(C_1) \setminus \{d(\bar{x} + y), d(\bar{x} + y)\}$$

for some i, and consequently $C_1 \gg C_2$.

(Invariance) First notice that by application of the merge rules – $(MLBi)$ and $(MUBi)$ – we can ensure that for each $\bar{x} + y$, C contains at most the inequalities

$$\begin{array}{ll}(a)\ c_1^- \preccurlyeq_1^- \bar{x} + y \preccurlyeq_1^+ c_1^+ & (c)\ c_y^- \preccurlyeq_y^- y \preccurlyeq_y^+ c_y^+ \\ (b)\ c_2^- \preccurlyeq_2^- \bar{x} + z \preccurlyeq_2^+ c_2^+ & (d)\ c_z^- \preccurlyeq_z^- z \preccurlyeq_z^+ c_z^+ \\ & (e)\ z + b \preccurlyeq_3^- y \preccurlyeq_3^+ z' + b'\end{array}$$

such that elimination of y according to Fourier's method can be reduced to their pairwise combinations. It can be checked that combination of (a) and (c) yields

$$c_1^- - \bar{x} \preccurlyeq_1^- \downarrow \preccurlyeq_y^+ c_y^+ \leftrightarrow c_1^- - c_y^+ \preccurlyeq_1^- \downarrow \preccurlyeq_y^+ \bar{x} \tag{14}$$

and

$$c_y^- \preccurlyeq_y^- \downarrow \preccurlyeq_1^+ c_1^+ - \bar{x} \leftrightarrow \bar{x} \preccurlyeq_y^- \downarrow \preccurlyeq_1^+ c_1^+ - c_y^-, \tag{15}$$

which correspond to applications of (*ILB*) and (*IUB*).

The combination of (a) and (e) leads to

$$c_1^- - \bar{x} \preccurlyeq_1^- \downarrow \preccurlyeq_3^+ z' + b' \leftrightarrow c_1^- - b' \preccurlyeq_1^- \downarrow \preccurlyeq_3^+ \bar{x} + z'$$
$$z + b \preccurlyeq_1^+ \downarrow \preccurlyeq_3^- c_1^+ - \bar{x} \leftrightarrow \bar{x} + z \preccurlyeq_1^+ \downarrow \preccurlyeq_3^- c_1^+ - b$$

and that of (c) and (e) to

$$c_y^- \preccurlyeq_y^- \downarrow \preccurlyeq_3^+ z' + b' \leftrightarrow c_y^- - b' \preccurlyeq_y^- \downarrow \preccurlyeq_3^+ z'$$
$$z + b \preccurlyeq_3^- \downarrow \preccurlyeq_y^+ c_y^+ \leftrightarrow z \preccurlyeq_3^- \downarrow \preccurlyeq_y^+ c_y^+ - b.$$

These inequalities correspond to applications of the (*Eyi*)-rules. The premises of the $\rightarrow_{edsc}$-simplification rules ensure that only inequalities not already subsumed are generated.

(Completeness) Since $\rightarrow_{edsc}$ specifies a particular variable elimination strategy of the Fourier's algorithm in conjunction with elimination of redundant inequalities, completeness follows by Lemma 7.4.

7.2. *Quantifier elimination*

The most important step towards a proof method for bounded universal Horn formulae exploiting uniform proofs are the following two lemmas. They allow to eliminate bounded universal quantifiers over linear inequalities.

Lemma 7.7 (LR-Lemma). *Let D be a system of linear inequalities over the rational or real numbers, and t^-, t^+ terms not containing the variable y. Then*

$$\forall y(t^- \preccurlyeq^- y \preccurlyeq^+ t^+ \rightarrow \exists \mathbf{z} D) \leftrightarrow EQ(\forall y(t^- \preccurlyeq^- y \preccurlyeq^+ t^+ \rightarrow \exists \mathbf{z} D)), \tag{16}$$

where EQ is defined in Fig. 8.

Proof. Let α be a variable assignment for the variables in $Var(t^-)$ and $Var(t^+)$, and let $\bar{\alpha}(t^-) = c^-$, $\bar{\alpha}(t^+) = c^+$, and $\bar{\alpha}(D) = C$.

If $c^- \preccurlyeq^- \downarrow \preccurlyeq^+ c^+$ does not hold, then the left side of the equivalence (16) under α is trivially true, i.e. it is equivalent to $\neg(c^- \preccurlyeq^- \downarrow \preccurlyeq^+ c^+)$. If $c^- \preccurlyeq^- \downarrow \preccurlyeq^+ c^+$ hold, then using the equivalence

$$\forall y(c^- \preccurlyeq^- y \preccurlyeq^+ c^+ \rightarrow \exists \mathbf{z} C) \leftrightarrow \bigwedge_{c^- \preccurlyeq^- c \preccurlyeq^+ c^+} \exists \mathbf{z}(\underbrace{C \cup \{c \leqslant y_c \leqslant c\}}_{C_c})$$

Function $EQ(\forall y(t^- \preccurlyeq^- y \preccurlyeq^+ t^+ \rightarrow \exists \mathbf{z} C))$

Input: *A set of linear inequalities* C *quantified* $\forall y(t^- \preccurlyeq^- y \preccurlyeq^+ t^+ \rightarrow \exists \mathbf{z} C)$, *terms* t^-, t^+ *with* $y \notin Var(t^-) \cup Var(t^+)$, *— i.e., y is not contained in the variables of* t^-, t^+ *—,* $\mathbf{z}$ *tuples of variables,* $\preccurlyeq^-, \preccurlyeq^+ \in \{\leqslant, <\}$.

Output: *A to the input formula equivalent quantifier free formula* C'.

begin

Eliminate $\mathbf{z}$ *from* $C \cup \{t^- \preccurlyeq^- y \preccurlyeq^+ t^+\}$ *via the Fourier's method. Arrange the result into a form suitable for the elimination of* y, *i.e., into*

$$C_1 = \{ \begin{array}{lll} l_i \preccurlyeq_i^l & y & i = 1, \ldots, p \\ & y \preccurlyeq_j^r r_j & j = 1, \ldots, q \\ & d_i \preccurlyeq_i^d 0 & i = 1, \ldots, s\}. \end{array}$$

$$C_2 = \{t^- \preccurlyeq^- \downarrow \preccurlyeq^+ t^+\} \cup \{d_i \preccurlyeq_i^d 0 \mid i = 1, \ldots, s\}$$

$$C_l = \textbf{if } \preccurlyeq^- = < \textbf{ then } \bigcup_{i=1}^{p} \{l_i \leqslant t^-\} \quad \textbf{else } \bigcup_{i=1}^{p} \{l_i \preccurlyeq_i^l t^-\}$$

$$C_r = \textbf{if } \preccurlyeq^+ = < \textbf{ then } \bigcup_{j=1}^{q} \{t^+ \leqslant r_j\} \quad \textbf{else } \bigcup_{j=1}^{q} \{t^+ \preccurlyeq_j^+ r_j\}$$

return $(C_2 \cup C_l \cup C_r) \vee \{\neg(t^- \preccurlyeq^- \downarrow \preccurlyeq^+ t^+)\}$

end

Fig. 8. Elimination of bounded $\forall$ and $\exists$ quantifiers.

we eliminate $\mathbf{z}$ via Fourier's methods and obtain sets C_c'. We arrange them in a form suitable for the elimination of y_c:

$$l_i \preccurlyeq_i^l y_c \quad i = 1, \ldots, p$$

$$c \leqslant y_c$$

$$y_c \preccurlyeq_j^r r_j \quad j = 1, \ldots, q$$

$$y_c \leqslant c$$

$$d_i \preccurlyeq_i^d 0 \quad i = 1, \ldots, s.$$

Elimination of y_c yield sets C_c'' with $C_c'' =$

$$l_i \preccurlyeq_i^l \downarrow \preccurlyeq_j^r r_j \quad i = 1, \ldots, p; \; j = 1, \ldots, q$$

$$l_i \preccurlyeq_i^l c \quad i = 1, \ldots, p$$

$$c \preccurlyeq_j^r r_j \quad j=1,\ldots,q$$
$$d_i \preccurlyeq_i^d 0 \quad i=1,\ldots,s.$$

Their union $\bigcup_{c^- \preccurlyeq^- c \preccurlyeq^+ c^+} C_c''$ can then be written as

$$\{l_i \preccurlyeq_i^l \downarrow \preccurlyeq_j^r r_j \mid i=1,\ldots,p; j=1,\ldots,q\} \cup \{d_i \preccurlyeq_i^d 0 \mid i=1,\ldots,s\}$$
$$\cup \bigcup_{i=1}^{q} \{c \preccurlyeq_i^r r_i \mid c^- \preccurlyeq^- c \preccurlyeq^+ c^+\} \tag{17}$$
$$\cup \bigcup_{i=1}^{p} \{l_i \preccurlyeq_i^l c \mid c^- \preccurlyeq^- c \preccurlyeq^+ c^+\}. \tag{18}$$

The sets (17), (18) can be simplified, since for $\preccurlyeq^+ = <$, $c^+ \leqslant r_i \rightarrow c \preccurlyeq_i^r r_i$ for all $c < c^+$. On the other hand, for each solution α of $S = \{c \preccurlyeq_i^r r_i \mid c^- \preccurlyeq^- c < c^+\}$: $c^+ \leqslant \overline{\alpha}(r_i)$, since in case of $\overline{\alpha}(r_i) < c^+$, there would be a c_α with $\overline{\alpha}(r_i) < c_\alpha < c^+$ (Density of $\mathbb{Q}$ and $\mathbb{R}$), in contradiction to $c_\alpha \preccurlyeq_i^r \overline{\alpha}(r_i)$, since $\{c_\alpha \preccurlyeq_i^r r_i\} \subseteq S$. Thus, $S \leftrightarrow c^+ \leqslant r_i$ and thereby

$$(17) \leftrightarrow \bigcup_{i=1}^{q} \{c^+ \leqslant r_i\}.$$

For $\preccurlyeq^+ = \leqslant$, notice that

$$c^+ \preccurlyeq_i^r r_i \rightarrow \{c \preccurlyeq_i^r r_i \mid c^- \preccurlyeq^- c \leqslant c^+\}$$

holds and because of $c^+ \preccurlyeq_i^r r_i \in \{c \preccurlyeq_i^r r_i \mid c^- \preccurlyeq^- c \leqslant c^+\}$ also the inverse implication such that

$$(17) \leftrightarrow \bigcup_{i=1}^{q} \{c^+ \preccurlyeq_i^r r_i\}.$$

Since by assumption $c^- \leqslant c^+$,

$$(l_i \preccurlyeq_i^l c^- \wedge c^+ \preccurlyeq_j^r r_j) \rightarrow l_i \preccurlyeq_i^l \downarrow \preccurlyeq_j^r r_j \quad i=1,\ldots,p;\ j=1,\ldots,q.$$

Consequently,

$$\{l_i \preccurlyeq_i^l \downarrow \preccurlyeq_j^r r_j \mid i=1,\ldots,p; j=1,\ldots,q\} \cup \{d_i \preccurlyeq_i^d 0 \mid i=1,\ldots,s\}$$
$$\cup \bigcup_{i=1}^{q} \{c \preccurlyeq_i^r r_i \mid c^- \preccurlyeq^- c \preccurlyeq^+ c^+\}$$
$$\cup \bigcup_{i=1}^{p} \{l_i \preccurlyeq_i^l c \mid c^- \preccurlyeq^- c \preccurlyeq^+ c^+\}$$
$$\leftrightarrow$$
$$\{d_i \preccurlyeq_i^d 0 \mid i=1,\ldots,s\} \cup \bigcup_{i=1}^{q} \{c \preccurlyeq_i^r r_i \mid c^- \preccurlyeq^- c \preccurlyeq^+ c^+\}$$
$$\cup \bigcup_{i=1}^{p} \{l_i \preccurlyeq_i^l c \mid c^- \preccurlyeq^- c \preccurlyeq^+ c^+\},$$

and therefore

$$\forall y(c^- \preccurlyeq^- y \preccurlyeq^+ c^+ \to \exists \mathbf{z} C)$$
$$\leftrightarrow$$
$$EQ(\forall y(c^- \preccurlyeq^- y \preccurlyeq^+ c^+ \to \exists \mathbf{z} C))$$

and since $\bar{\alpha}(t^-) = c^-, \bar{\alpha}(t^+) = c^+$, and $\bar{\alpha}(D) = C$ also (16). □

Lemma 7.8 (LR-Lemma for ETCSs over the integers). *If C is an extended tree constraint system over the integers and t^-, t^+ terms not containing y, then*

$$\forall y(t^- \leqslant y \leqslant t^+ \to \exists \mathbf{z} C) \leftrightarrow ((\exists \bar{z} C\{y \leftarrow t^-\} \wedge \exists \mathbf{z} C\{y \leftarrow t^+\} \wedge t^- \leqslant t^+) \vee t^- > t^+).$$

Remark. The lemma does not hold for arbitrary set of inequalities over the integers, e.g., $\forall y(0 \leqslant y \leqslant 3 \to \exists x(0 \leqslant 3x - y \leqslant 0))$ is false but $\exists x((0 \leqslant 3x - y \leqslant 0)\{y \leftarrow 0\}) \wedge \exists x((0 \leqslant 3x - y \leqslant 0)\{y \leftarrow 3\})$ is true over the integers.

Proof (*sketch*). Follow the lines of the proof of the LR-lemma for systems of inequalities over the reals or rational numbers. The only crucial point in the proof depending on the structure of rational or reals – restricting the lemma to inequalities $\leqslant$ – is the variable elimination by Fourier's method. Following the variable elimination strategy used in $\to_{esc}$ for extended tree constraint systems we eliminate every variable without algebraic operations on its coefficient. Therefore, the proof goes through for extended tree constraint systems over the integers as well. □

The union of two ETCSs (with respect to a common tree) can be again represented as an ETCS.

Lemma 7.9 ($\bigcup$-Lemma). *Let P be an extended CLP-program and G an extended CLP-goal. If*

$$P \models \tilde{\forall} I_1 \to (\forall y(c_1^- \preccurlyeq_1^- y \preccurlyeq_1^+ c_1^+ \to G))$$
$$P \models \tilde{\forall} I_2 \to (\forall y(c_2^- \preccurlyeq_2^- y \preccurlyeq_2^+ c_2^+ \to G))$$

with $c_i^- \leqslant c_i^+$ and either

1. $c_1^+ \geqslant c_2^-$, $\preccurlyeq_1^+ = \leqslant$ *or* $\preccurlyeq_2^- = \leqslant$, *or*
2. $c_1^+ > c_2^-$,

then $P \models \tilde{\forall} I_1 \cup I_2 \to (\forall y(c_1^- \preccurlyeq_1^- y \preccurlyeq_2^+ c_2^+ \to G))$, where $\tilde{\forall}$ denotes the universal closure.

In the discrete case the union of two ETCSs can be also represented without strict inequalities.

Lemma 7.10 ($\bigcup$-Lemma). *Let P be an extended CLP-program and G be an extended CLP-goal over $\mathscr{A}_1$. If*

1. $P \models \tilde{\forall}(I_1 \to (\forall y(c_1^- \leqslant y \leqslant c_1^+ \to G)))$ *and*
2. $P \models \tilde{\forall}(I_2 \to (\forall y(c_2^- \leqslant y \leqslant c_2^+ \to G)))$,

with $c_i^- \leqslant c_i^+$, $i=1,2$, *and* $c_1^+ \geqslant c_2^- - 1$, *then* $P \models \tilde{\forall}(I_1 \cup I_2 \rightarrow (\forall y(c_1^- \leqslant y \leqslant c_2^+ \rightarrow G)))$, *where* $\tilde{\forall}$ *denotes the universal closure.*

8. Proof method, discrete time

LR-lemma and $\bigcup$-lemma are the key of our proof method for proving bounded universal goals. In Fig. 9 we extend simple MTL-resolution to prove goals containing $\Box_c$ operators. $G[A]$ indicates that A occurs in G, and then $G[B]$ denotes the result of replacing one occurrence of A in G by B. The $\forall I$ rules generate existential relaxations of the universal goals to be proven. The sets of answer constraints obtained by successful derivations of the existential goals, are converted by $\exists\forall$-C-rule into answer constraints of bounded universal goals by application of the LR-lemma and combined by the $\forall C$-rule ($\bigcup$-lemma). $\forall E$ rules check whether the solution set of the set of constraints obtained subsumes the universally quantified goal to be proven.

Goals containing $\mathscr{S}$ and $\mathscr{U}$ operators can be handled essentially in the same way as goals containing $\Box_c$ operators (Fig. 11). First, we prove the existential part of the translated $A\,\mathscr{S}\,B$ goal (respectively, $A\,\mathscr{U}\,B$ goal) and then the bounded universal part by the proof method for $\Box_c$ goals. We refine the translation function in Fig. 10 in order to avoid strict inequalities and to mark $\wedge$ operators introduced by translation of $\mathscr{S}$ and $\mathscr{U}$. We use these labels to identify the existential and the universal part in the translated goals and to control the sequence of their proofs (Fig. 11). The derivation relation defined by rules listed in Figs. 11 and 9, and the MTL-resolution rule, Fig. 4, in conjunction with satisfiability checking over $\mathscr{A}_1$ is denoted by $\vdash_{EMTL}$.

Lemma 8.1. *Let* $C_0 \wedge G_0$ *be an extended CLP-goal. If* $C_0 \wedge G_0 \vdash_{EMTL} C_1 \wedge G_1$ *with respect to an extended CLP-program* P*, then* C_1 *is an extended tree constraint system.*

The completeness of the EMTL-calculus is essentially ensured by the following lemma which allows to show the continuity of the $T_{(\Pi(P),\mathscr{A}_1)}$-operator.

Lemma 8.2. *Let* $I_1 \subseteq I_2 \subseteq \ldots$ *be an ascending chain of* $\mathscr{A}_1$*-structures and* G *an extended CLP-goal. If* $\bigcup_{i=1}^{\infty} I_i \models G$*, then there is a natural number* j *with* $I_j \models G$.

As a direct consequence, we obtain:

Lemma 8.3. $T_{(P,\mathscr{A}_1)}$ *is continuous, i.e.,* $T_{(P,\mathscr{A}_1)}(\bigcup_{i=1}^{\infty} I_i) = \bigcup_{i=1}^{\infty} T_{(P,\mathscr{A}_1)}(I_i)$ *for each ascending chain* $I_1 \subseteq I_2 \subseteq \ldots$

The operator $T_{(\Pi(P),\mathscr{A}_2)}$ is, however, not continuous in general as we will see in the next section. We can now prove soundness and completeness of the $\vdash_{EMTL}$ calculus.

$$(\forall I)\quad \frac{G[\forall y\,(\{c^- \leqslant y \leqslant c^+\} \to \pi(B,\bar{x}+y))]}{G[\forall y\,(\{c^- \leqslant y \leqslant c^+\} \to \pi(B,\bar{x}+y) \mid)],}$$

$$(\exists I)\quad \frac{G[\forall y\,(\{t^- \leqslant y \leqslant t^+\} \to \pi(B,\bar{x}+y) \mid \bigwedge_i B^i_y)]}{\begin{array}{l} G[\forall y\,(\{t^- \leqslant y \leqslant t^+\} \to \pi(B,\bar{x}+y) \mid \\ (\pi(B,\bar{x}+y) \wedge \{t^- \leqslant y \leqslant t^+\})\theta \wedge \bigwedge_i B^i_y)]\end{array}}$$

if θ is a permutation on integer variables, i.e., a bijective variable renaming that maps the integer variables in $Var(\pi(B,\bar{x}+y)) \setminus \{\bar{x}\}$ to new distinct variables.

$$(\exists\forall C)\quad \frac{\begin{array}{l}\leftarrow G[\forall y\,(\{t^- \leqslant y \leqslant t^+\} \to \pi(B,\bar{x}+y) \mid \\ \{c_1^- \leqslant y_1 \leqslant c_1^+\} \cup I_1 \wedge \bigwedge_i B^i_y)]\end{array}}{\begin{array}{l}\leftarrow G[\forall y\,(\{t^- \leqslant y \leqslant t^+\} \to \pi(B,\bar{x}+y) \mid \\ \bigcup_{\pi=\theta,\sigma} (\{c_1^- \leqslant y_1 \leqslant c_1^+\} \cup I_1)\pi \cup \{y\theta \leqslant y\sigma\} \wedge \bigwedge_i B^i_y)],\end{array}}$$

where θ, σ are permutations on variables in $Var(I_1 \cup \{c_1^- \leqslant y_1 \leqslant c_1^+\} \setminus \{\bar{x}\})$ that map the variables in their domains to new distinct variables.

$$(\forall C)\quad \frac{\begin{array}{l}\leftarrow G[\forall y\,(\{t^- \leqslant y \leqslant t^+\} \to \pi(B,\bar{x}+y) \mid \\ \bigcup_{i=1,2} (I_i \cup \{y\theta_i \leqslant y\sigma_i\}) \wedge \bigwedge_i B^i_y)]\end{array}}{\begin{array}{l}\leftarrow G[\forall y\,(\{t^- \leqslant y \leqslant t^+\} \to \pi(B,\bar{x}+y) \mid \bigcup_{i=1,2} (I_i \cup \{y\theta_i \leqslant y\sigma_i\}) \\ \cup \{y\theta_1 \leqslant y\theta_2, y\theta_2 - 1 \leqslant y\sigma_1 \leqslant y\sigma_2\} \wedge \bigwedge_i B^i_y)]\end{array}}$$

$$(\forall E1)\quad \frac{\leftarrow G[\forall y\,(\{t^- \leqslant y \leqslant t^+\} \to \pi(B,\bar{x}+y) \mid \bigwedge_i B^i_y)]}{\leftarrow G[\{t^- \geqslant t^+ - 1\}]}$$

if $t^- > t^+$ is satisfiable.

$$(\forall E2)\quad \frac{\leftarrow G[\forall y\,(\{t^- \leqslant y \leqslant t^+\} \to \pi(B,\bar{x}+y) \mid I \cup \{y\theta \leqslant y\sigma\} \wedge \bigwedge_i B^i_y)]}{\leftarrow G[I \cup \{y\theta \leqslant y\sigma\} \cup \{y\theta \leqslant t^- \leqslant t^+ \leqslant y\sigma\}]}$$

if $I \cup \{y\theta \leqslant y\sigma\} \cup \{y\theta \leqslant t^- \leqslant t^+ \leqslant y\sigma\}$ is satisfiable.

Fig. 9. Extended MTL-resolution for proving $\Box_c$-goals.

Theorem 8.4. *Let P' be a set of bounded universal Horn formulae, G' a bounded universal goal, and P (respectively, G) its translation.*

(Soundness) *If $P \vdash_{EMTL} G$ then $P' \models G'$,*

(Completeness) *If $P' \models G'$ then $P \vdash_{EMTL} G$.*

$$\pi(A\,\mathcal{S}_c\,B,t)=\exists y_s((\{c\leqslant y_s\leqslant -1\}\wedge\pi(B,t+y_s))\dot\wedge$$
$$\forall y_s'(\{y_s+1\leqslant y_s'\leqslant -1\}\rightarrow\pi(A,t+y_s')))$$
$$\pi(A\,\mathcal{U}_c\,B,t)=\exists y_u((\{1\leqslant y_u\leqslant c\}\wedge\pi(B,t+y_u))\dot\wedge$$
$$\forall y_u'(\{1\leqslant y_u'\leqslant y_u-1\}\rightarrow\pi(A,t+y_u')))$$

Fig. 10. Translation of $\mathcal{S}$ and $\mathcal{U}$ goals for discrete time.

$$(\forall I\mathcal{S})\frac{\leftarrow G[(I\cup\{c\leqslant y\leqslant d\}\dot\wedge\forall y'\ (\{y+1\leqslant y'\leqslant c^+\}\rightarrow\pi(B,\bar x+y')))]}{\leftarrow G[I\cup\{c\leqslant y\leqslant d\}\wedge\forall y'\ (\{y+1\leqslant y'\leqslant c^+\}\rightarrow\pi(B,\bar x+y')\mid)]}$$

$$(\forall I\mathcal{U})\frac{\leftarrow G[(I\cup\{c\leqslant y\leqslant d\}\dot\wedge\forall y'(\{c^-\leqslant y'\leqslant y-1\}\rightarrow\pi(B,\bar x+y')))]}{\leftarrow G[I\cup\{c\leqslant y\leqslant d\}\wedge\forall y'(\{c^-\leqslant y'\leqslant y-1\}\rightarrow\pi(B,\bar x+y')\mid)]}$$

Fig. 11. Extended MTL-resolution for $\mathcal{S}$ and $\mathcal{U}$ goals.

Proof. Due to soundness and completeness of the translation, Proposition 4.1, it remains to show their counterparts for extended CLP-programs and goals.

(*Soundness*) We prove

$$G\vdash_{EMTL} I\Rightarrow P\models\tilde\forall(I\rightarrow G),\tag{19}$$

by induction on the number n of $\forall I$ rule applications, which shows the soundness part of the theorem.

For $n=0$, $G\vdash_{EMTL} I$ is a pure MTL-derivation and (19) follows by soundness of $(\Pi(P),\mathcal{A}_1)$-derivations.

For $n>0$, $G\vdash_{EMTL} I$ is of the form[11]

$$\begin{aligned}G\ \vdash_{MTL}\quad & G'=G'[\forall y(t^-\leqslant y\leqslant t^+\rightarrow B(\bar x+y))]\\ \vdash_{\forall I}\quad & G'[\forall y(t^-\leqslant y\leqslant t^+\rightarrow B(\bar x+y))\mid]\\ \vdash_{EMTL}\quad & G'[\forall y(t^-\leqslant y\leqslant t^+\rightarrow B(\bar x+y))\mid\textstyle\bigwedge B_i(\bar x+y)\theta_i]\end{aligned}$$

[11] If $G\vdash\cdots\vdash_{\forall E1} G''$, soundness follows directly by inductive hypothesis.

$$\begin{array}{ll}
\vdash_{EMTL} & G'[\forall y(t^- \leqslant y \leqslant t^+ \to B(\bar{x}+y)) \mid \bigwedge I_i] \\
\vdash_{EMTL} & G'[\forall y(t^- \leqslant y \leqslant t^+ \to B(\bar{x}+y)) \mid \bigwedge \underbrace{I_i\theta_i \cup I_i\sigma_i \cup \{y\theta_i \leqslant y\sigma_i\}}_{I^i}] \\
\vdash_{EMTL} & G'[\forall y(t^- \leqslant y \leqslant t^+ \to B(\bar{x}+y)) \mid \\
& \underbrace{\bigwedge I^i \wedge \bigcup_{i=1}^{n} \{y_i\theta_i \leqslant y_{i+1}\theta_{i+1}, y_{i+1}\theta_{i+1} - 1 \leqslant y_i\sigma_i \leqslant y_{i+1}\sigma_{i+1}\}]}_{I'} \\
\vdash_{\forall E2} & G'' = G'[S \cup \{y_1\theta_1 \leqslant t^-, t^+ \leqslant y_{n+1}\sigma_{n+1}\}] \\
\vdash_{MTL} & I.
\end{array}$$

Applying the inductive hypothesis,

$$P \models \tilde{\forall}(I_i \to B(\bar{x}+y)\theta_i) \tag{20}$$

and since for each $c^-, c^+ \in \mathbb{Z}$ with $c^- \leqslant c^+$ according to the LR-lemma, Lemma 7.8,

$$[\![\forall y(\{c^- \leqslant y \leqslant c^+\} \to \exists \mathbf{z} I_i)]\!] = [\![I^i\{y\theta_i \leftarrow c^-, y\sigma_i \leftarrow c^+\}]\!]_{Var(I_i)\setminus dom(\theta_i)},$$

also

$$P \models \tilde{\forall}(I^i \to (\forall y(y\theta_i \leqslant y \leqslant y\sigma_i \to \exists \mathbf{z} B(\bar{x}+y)))). \tag{21}$$

Using $\bigcup$-Lemma for $i = 1,\dots,n$, we obtain

$$P \models \tilde{\forall}(I' \to (\forall y(y\theta_1 \leqslant y \leqslant y\sigma_{n+1} \to \exists \mathbf{z} B(\bar{x}+y)))) \tag{22}$$

and again by LR-lemma,

$$P \models \tilde{\forall}((\underbrace{S \cup \{y_1\theta_1 \leqslant t^-, t^+ \leqslant y_{n+1}\sigma_{n+1}\}}_{S'}) \to \forall y(t^- \leqslant y \leqslant t^+ \to \exists \mathbf{z} B(\bar{x}+y))). \tag{23}$$

By inductive hypothesis, $P \models \tilde{\forall}(I \to G'')$ such that with (23) and $I \supseteq S'$ also $P \models \tilde{\forall}(I \to G'' \to G')$ implying $P \models \tilde{\forall}(I \to G')$. Using soundness of $(\Pi(P), \mathscr{A}_1)$-derivations, $P \models \tilde{\forall}(I \to (G' \to G))$ and consequently $P \models \tilde{\forall}(I \to G)$.

(*Completeness*) According to $\mathscr{A}$-model-lemma and continuity of $T_{(\Pi(P),\mathscr{A}_1)}$,

$$lm(\Pi(P), \mathscr{A}_1) = lfp(T_{(\Pi(P),\mathscr{A}_1)}) = T_{(\Pi(P),\mathscr{A}_1)} \uparrow \omega,$$

and it remains to be shown: If $T_{(P,\mathscr{A}_1)} \uparrow n \models G\theta$ for some θ, then $\tilde{\exists} G \vdash_{EMTL} I$ with $\theta \in [\![I]\!]$, which we are going to prove by induction on n.

The base case, $n = 0$, is trivial. For $n > 0$, we continue by structural induction on G. The cases $G = A$, $G = G_1 \wedge G_2$, and $\exists x G$ are shown by standard arguments. For G being of the form $G = \forall y(c^- \leqslant y \leqslant c^+ \to G')$,

$$T_{(P,\mathscr{A}_1)} \uparrow n \models \forall y(c^- \leqslant y \leqslant c^+ \to G')\theta$$

if $T_{(P,\mathscr{A}_1)} \uparrow n \models \{c \leqslant y \leqslant c\} \wedge G'\theta$ for all c within $c^- \leqslant c \leqslant c^+$. Inductive hypothesis applied to $\{c \leqslant y \leqslant c\} \wedge G'\theta$ yields the existence of derivations $\{c \leqslant y \leqslant c\} \wedge G'\theta \vdash_{EMTL}$

I_c with $\theta\{y \leftarrow c\} \in [\![I_c]\!]_{Var(G')}$ and $\{c^- \leqslant y\theta_c \leqslant c^+\} \wedge G'\theta\theta_c \vdash_{EMTL} I'_c$ with $I'_c \subseteq I_c\theta_c$ for some permutations θ_c. Since

$$[\![I'_c]\!]_{Var(I'_c)} = [\![\underbrace{I'_c \cup I'_c\{y\theta_c \leftarrow y\sigma_c\} \cup \{y\theta_c \leqslant y\sigma_c\}}_{I''_c}]\!]_{Var(I'_c)}$$

we have for

$$I^{\forall C} = \bigcup_{c=c^-}^{c^+-1} (I''_c \cup \{y\theta_c \leqslant y\theta_{c+1}, y\theta_{c+1} - 1 \leqslant y\sigma_c \leqslant y\sigma_{c+1}\}) \cup I''_{c^+},$$

$$[\![I^{\forall C}]\!]_{Var(G)} \supseteq \left[\!\!\left[\bigcup_{c=c^-}^{c^+} I_c\right]\!\!\right]_{Var(G)},$$

and, as a consequence, an EMTL-derivation

$$\begin{array}{lll} \tilde{\exists} G & \vdash_{EMTL} & \\ & \vdots & \\ & \vdash_{\forall C} & \forall y(c^- \leqslant y \leqslant c^+ \rightarrow G') \mid I^{\forall C}. \end{array}$$

On the other hand,

$$[\![I_c\theta_c]\!] = [\![I''_c \cup \{c \leqslant y\theta_c \leqslant c, c \leqslant y\sigma_c \leqslant c\}]\!]_{Var(I_c\theta_c)}$$

implies

$$\begin{array}{c} [\![\underbrace{I^{\forall C} \cup \{y\theta_{c^-} \leqslant c^-, c^+ \leqslant y\sigma_{c^+}\}}_{I}]\!]_{Var(\bigcup I_c\theta_c)} \\ \supseteq \\ \left[\!\!\left[\bigcup_{c=c^-}^{c^+} I_c\theta_c\right]\!\!\right]_{Var(\bigcup I_c\theta_c)} \end{array} \tag{24}$$

and the derivation above can be extended by $\vdash_{\forall E2}$ I. Since $\theta \in [\![I_c]\!]_{Var(G)}$ for all c within $c^- \leqslant c \leqslant c^+$, (24) implies $\theta \in [\![I]\!]_{Var(G)}$, which concludes this case.

For G being of the form $\exists x(G(x) \dot{\wedge} \forall y(t^- + x \leqslant y \leqslant t^+ \rightarrow G'))$,

$$T_{(P,\mathscr{A}_1)} \uparrow n \models \exists x(G(x) \dot{\wedge} \forall y(t^- + x \leqslant y \leqslant t^+ \rightarrow G'))\theta$$

implies that there is an α with $\alpha(x) \in \mathbb{Z}$ such that

$$T_{(P,\mathscr{A}_1)} \uparrow n \models \underbrace{(G(x) \dot{\wedge} \forall y(t^- + x \leqslant y \leqslant t^+ \rightarrow G'))\theta\alpha}_{G(x)\theta\alpha \dot{\wedge} \forall y(t^- + \alpha(x) \leqslant y \leqslant t^+ \rightarrow G'\theta)}.$$

Since $t^- + \alpha(x) \in \mathbb{Z}$, according to inductive hypothesis applied to $\forall y(t^- + \alpha(x) \leqslant y \leqslant t^+ \rightarrow G')$, there is a derivation

$$G(x)\alpha \dot{\wedge} \forall y(t^- + \alpha(x) \leqslant y \leqslant t^+ \rightarrow G) \vdash_{EMTL} I$$

$$
\begin{array}{ll}
 & \leftarrow \text{manager}(X)\,\mathscr{S}\,\text{salesman}(X) \equiv_{\Pi} \leftarrow \exists y(\{y \leqslant -1\} \wedge \text{salesman}(y, X) \\
 & \qquad \dot{\wedge} \forall y'(\{y+1 \leqslant y' \leqslant -1\} \rightarrow \text{manager}(y', X))) \\
\vdash_{MTL} & \{-20 \leqslant y \leqslant -16\} \dot{\wedge} \forall y'(\{y+1 \leqslant y' \leqslant -1\} \rightarrow \text{manager}(y', \text{john})) \\
\vdash_{\forall I\,\mathscr{S}} & \{-20 \leqslant y \leqslant -16\} \wedge \forall y'(\{y+1 \leqslant y' \leqslant -1\} \rightarrow \text{manager}(y', \text{john}) \mid) \\
\vdash_{\exists I} & \{-20 \leqslant y \leqslant -16\} \wedge \forall y'(\{y+1 \leqslant y' \leqslant -1\} \rightarrow \text{manager}(y', \text{john}) \mid \\
 & \{y+1 \leqslant y'_1 \leqslant -1\} \wedge \text{manager}(y'_1, \text{john}) \wedge \\
 & \{y+1 \leqslant y'_2 \leqslant -1\} \wedge \text{manager}(y'_2, \text{john}) \wedge \\
 & \{y+1 \leqslant y'_3 \leqslant -1\} \wedge \text{manager}(y'_3, \text{john})) \\
\vdash_{MTL} & \{-20 \leqslant y \leqslant -16\} \wedge \forall y'(\{y+1 \leqslant y' \leqslant -1\} \rightarrow \text{manager}(y', \text{john}) \mid \\
 & \{-15 \leqslant y'_1 \leqslant -11\} \wedge \{-10 \leqslant y'_2 \leqslant -6\} \wedge \{-5 \leqslant y'_3 \leqslant -1\}) \\
\vdash_{\exists\forall C} & \{-20 \leqslant y \leqslant -16\} \wedge \forall y'(\{y+1 \leqslant y' \leqslant -1\} \rightarrow \text{manager}(y', \text{john}) \mid \\
 & \underbrace{\begin{array}{l} \bigcup_{\pi=\theta,\sigma}(\{-15 \leqslant y'_1 \leqslant -11\})\pi \cup \{y'_1\theta \leqslant y'_1\sigma\} \wedge \\ \bigcup_{\pi=\theta,\sigma}(\{-10 \leqslant y'_2 \leqslant -6\})\pi \cup \{y'_2\theta \leqslant y'_2\sigma\} \wedge \\ \bigcup_{\pi=\theta,\sigma}(\{-5 \leqslant y'_3 \leqslant -1\})\pi \cup \{y'_3\theta \leqslant y'_3\sigma\}) \end{array}}_{I} \\
\vdash_{\forall C} & \{-20 \leqslant y \leqslant -16\} \wedge \forall y'(\{y+1 \leqslant y' \leqslant -1\} \rightarrow \text{manager}(y', \text{john}) \mid \\
 & \underbrace{\begin{array}{l} I \cup \{y'_1\theta \leqslant y'_2\theta, y'_2\theta - 1 \leqslant y'_1\sigma \leqslant y'_2\sigma\} \\ \quad \cup \{y'_2\theta \leqslant y'_3\theta, y'_3\theta - 1 \leqslant y'_2\sigma \leqslant y'_3\sigma\}) \end{array}}_{I_1 \Rightarrow y'_2\theta = -10,\, y'_1\sigma = -11,\, y'_3\theta = -5,\, y'_2\sigma = -6} \\
\vdash_{\forall E2} & \{-20 \leqslant y \leqslant -16\} \cup I_1 \cup \{y'_1\theta \leqslant y+1 \leqslant -1 \leqslant y'_3\sigma\} \\
 & \Rightarrow y = -16,\, y'_1\theta = -15,\, y'_3\sigma = -1
\end{array}
$$

Fig. 12. An EMTL-derivation for the goal from Example 8.5.

with $\theta \in [\![I]\!]_{Var(G)}$, and, as a consequence, also one of the form

$$G(x)\dot{\wedge}\forall y(t^- + x \leqslant y \leqslant t^+ \rightarrow G') \vdash_{EMTL} I^*$$

consisting of the same derivations steps. We have $[\![I^*]\!] \supseteq [\![I]\!]$ and $\theta \in [\![I^*]\!]_{Var(G)}$. □

Example 8.5. Consider again the program from the Example 7.3 and the goal

$$\leftarrow \text{manager}(X)\ \mathscr{S}\ \text{salesman}(X) \equiv_{\Pi} \leftarrow \exists y(\{y \leqslant -1\} \wedge \text{salesman}(y,X)$$
$$\dot{\wedge}\forall y'(\{y+1 \leqslant y' \leqslant -1\} \rightarrow \text{manager}(y',X))).$$

It can be proved by the $\vdash_{EMTL}$-derivation listed in Fig. 12.

Lemma 8.6. *Let C be a tree constraint system which is normalized with respect to the merge rules (MLBi) and (MUBi), i.e., (MLBi) and (MUBi) are not applicable to C.*

The satisfiability of C is decidable in $O(n)$ via $\rightarrow_{dsc}$, where n denotes the number of variables in C.

Proof. The number of variables in C corresponds to the number of nodes in $(Var(C), \leqslant)$. Since C is a tree constraint system none of the rules (Eyi) is applicable and by assumption also none of $(MLBi)$ and $(MUBi)$.

The rules (ILB) and (IUB) compute only improved bounds for strict prefixes of the applied inequalities $\bar{x} + y$. The normal form of C can therefore be computed applying rules (ILB) and (IUB) in a bottom up manner. Thus, if n denotes the number of nodes of the tree defined by C, then each of the rules (ILB) and (IUB) can be applied to C only n times. $\square$

Theorem 8.7. *Let P denote an extended CLP-program and $C \wedge G$ an extended CLP-goal.*

If $C \wedge G \vdash_{EMTL} C' \wedge G'$ and C is normalized with respect to $\rightarrow_{esc}$, then the satisfiability of C' is decidable in linear time via $\rightarrow_{esc}$.

Proof. Notice that if $G \vdash_{EMTL} G'$, then the tree underlying G' contains only a constant number (depending on the inference rule applied) of additional nodes and the corresponding ETCS C' a constant number of additional inequalities. The original ETCS C, that of G, is $\rightarrow_{esc}$-normalized and, as a consequence, $\rightarrow_{esc}$-simplification rules are only applicable to systems containing inequalities from $C' \backslash C$. It can be shown that for each of the EMTL rules C' can be (Eyi)-normalized by a constant number of simplification steps such that the theorem follows by Lemma 8.6. $\square$

9. Dense time structures

Temporal logics over dense time structures are in general harder than those over discrete time and methods developed for the discrete case usually do not adapt to the dense case directly. For example, $\square_{[c^-,c^+]}A \not\leftrightarrow \bigwedge_{i=1}^{n} \circ^i A$ for each $n \in \mathbb{N}$ and sets of bounded universal Horn formulae are not *compact* with respect to bounded universal goals, i.e. $X \models G$ iff for some finite subset $X' \subseteq X$, $X' \models G$. The later is reflected on the level of the $T_{(P,\mathscr{A}_2)}$ operator which is monotonic but not continuous (cf. Example 9.1). The validity problem is already Π_1^1-hard for most of the propositional real-time logics, including that of metric temporal logic underlying our work [7].

Nevertheless, the proof method presented in the preceding section can be shown, after slight modifications, to be complete for dense time also. This result, however, heavily relies on the fragment of bounded universal Horn formulae and requires to establish some basic results like an appropriate compactness property and approximation of the least model by at most ω steps of the $T_{(P,\mathscr{A}_2)}$ operator directly without recourse to standard methods.

Example 9.1. *Consider the chain*

$$I_0 \subseteq I_1 \subseteq I_2 \subseteq \dots$$

of MTL-structures defined by

$$T_{(P,\mathscr{A}_2)} \uparrow n = I_n$$

with respect to the program

$$\begin{aligned} P = \{ \ & p(x) \leftarrow 0.5 \leqslant x \leqslant 1, \\ & p(x) \leftarrow z \leqslant x \leqslant y \wedge y = z + z \wedge p(y), \\ & q(0) \leftarrow \forall y(\{0 < y \leqslant 1\} \rightarrow p(y)) \\ \} & \end{aligned}$$

We have

$$\begin{aligned} \bigcup_{i=0}^{\infty} I_i &= \bigcup_{i=0}^{\infty} T_{(P,\mathscr{A}_2)} \uparrow i \\ &= T_{(P,\mathscr{A}_2)} \uparrow \omega \not\models q(0), \end{aligned}$$

but because of

$$T_{(P,\mathscr{A}_2)} \uparrow n \models \forall x \left(\frac{1}{2^n} \leqslant x \leqslant 1 \rightarrow p(x) \right)$$

for $n > 0$ *and* $\lim_{n\to\infty} \frac{1}{2^n} = 0$, $T_{(P,\mathscr{A}_2)} \uparrow \omega \models \forall x(0 < x \leqslant 1 \rightarrow p(x))$, implying

$$\begin{aligned} & T_{(P,\mathscr{A}_2)}(T_{(P,\mathscr{A}_2)} \uparrow \omega) \models q(0), \ \text{but} \\ & \bigcup_{i=1}^{\infty} T_{(P,\mathscr{A}_2)} \uparrow n \not\models q(0). \end{aligned}$$

The chain $I_0 \subseteq I_1 \subseteq I_2 \subseteq \ldots$ can be generated by a set of bounded universal Horn formulae containing temporal operators with variables.

Example 9.2. A bounded universal Horn program containing temporal operators with variable bounds which results in a not continuous operator $T_{(\Pi(P),\mathscr{A}_2)}$.

$$\begin{aligned} & \Box_{[0.5,1]}\, p, \\ & \Box_{[x,1]}\, p \leftarrow \circ^x \circ^x p, \\ & q \leftarrow \Box_{(0,1]}\, p. \end{aligned}$$

In fact, proving bounded universal goals from programs with variable bounds forms already a Σ_1^1-complete problem (cf. Section 13).

10. Model presentation

An inspection of the Counterexample 9.1 shows that the chain $I_0 \subseteq I_1 \subseteq \ldots$ cannot be generated by a bounded universal Horn program containing temporal operators with constant bounds. In the following we restrict our attention to $\mathscr{A}_2$-models generated by programs without variable bounds.

Such models can be presented by atomic goals with constraints. An atomic goal $\leftarrow C \wedge p(\mathbf{r})$ presents a subset of the $\mathscr{A}_2$-base:

$$[\leftarrow C \wedge p(\mathbf{r})] = \{p(\mathbf{x})\theta \mid \mathscr{A} \models (C \cup \{\mathbf{x} = \mathbf{r}\})\theta\},$$

where $\mathbf{r}, \mathbf{x}$ denote tuples $r_1, \ldots, r_n$ (respectively, $x_1, \ldots, x_n$) and $\mathbf{x} = \mathbf{r}$, $x_1 = r_1, \ldots, x_n = r_n$; an arbitrary goal $\leftarrow C \wedge p_1(\mathbf{r}_1) \wedge \cdots \wedge p_n(\mathbf{r}_n)$ presents $[\leftarrow C \wedge p_1(\mathbf{r}_1) \wedge \cdots \wedge p_n(\mathbf{r}_n)]$, this is the set $\bigcup_{i=1}^{n} \{p_i(\mathbf{x}_i)\theta \mid \mathscr{A} \models (C \cup \{\mathbf{x}_1 = \mathbf{r}_1, \ldots, \mathbf{x}_n = \mathbf{r}_n\})\theta\}$. Given a set S of atomic constraints, $[S] = \bigcup_{\leftarrow C \wedge p(\mathbf{r}) \in S} [\leftarrow C \wedge p(\mathbf{r})]$. Such a S as given above is called *presentation*. A subset D of the $\mathscr{A}(\Sigma)$-basis is called *(finitely) presentable* if there is a (finite) presentation S such that $[S] = D$.

A presentation is called *tree presentation* if it is of the form $S = \bigcup_{i=1}^{n} \{C_i \wedge p(\bar{x}_i, \mathbf{x}_i)\}$ with $C_i = \{\bar{x}_i = x_{i1} + \cdots + x_{in_i} + b_i\} \cup C_i' \cup E_i$ such that C_i' are tree constraint systems with respect to a tree $(T_i, \leqslant)$, $x_{i1} + \cdots + x_{in_i} + b_i$ are paths in $(T_i, \leqslant)$ starting from the root, and E_i are term equations. For notational convenience, we often omit the term equations E_i and see tree presentations as sets $S = \bigcup_{i=1}^{n} \{\leftarrow C_i \wedge p_i(\bar{x}_i)\}$.

The operator $T_{(P,\mathscr{A}_2)}$ can be modeled on the level of presentations.

$$\begin{aligned} T_{(P,\mathscr{T})}(S) = \{\leftarrow C \wedge p(\mathbf{x}) \mid\ & \text{there is a Horn formula } p(\mathbf{r}) \leftarrow C_0 \wedge G \text{ in } P \text{ such} \\ & \text{that } S \models \tilde{\forall} C' \rightarrow G \text{ with } C' \text{ being } \mathscr{A}_2\text{-satisfiable} \\ & \text{and } C = C' \cup C_0 \cup \{\mathbf{r} = \mathbf{x}\}\}. \end{aligned}$$

$S \models \tilde{\forall} C \rightarrow G$ is defined by

1. $S \models \tilde{\forall} C \rightarrow p(\mathbf{r})$ for an atom $p(\mathbf{r})$ if there is a $\leftarrow C' \wedge p(\mathbf{x}) \in S$ with variables distinct to that of $\tilde{\forall} C \rightarrow p(\mathbf{r})$, and $C = C' \cup \{\mathbf{r} = \mathbf{x}\}$,
2. $S \models \tilde{\forall} C \rightarrow A \wedge B$ if $S \models \tilde{\forall} C_1 \rightarrow A$, $S \models \tilde{\forall} C_2 \rightarrow B$, and $C = C_1 \cup C_2$,
3. $S \models \tilde{\forall} C \rightarrow \exists x G$ if there are a quantifier free C and a C_1 such that $S \models \tilde{\forall} C_1 \rightarrow G$, and $\exists x C_1 \leftrightarrow C$,
4. $S \models \tilde{\forall} C \rightarrow \forall x(\{b^- \leqslant^- x \leqslant^+ b^+\} \rightarrow G)$ if there are C_b' such that $S \models \tilde{\forall} C_b' \rightarrow G$ and $\forall x(\{b^- \leqslant^- x \leqslant^+ b^+\} \rightarrow (\bigvee_{b^- \leqslant^- b \leqslant^+ b} C_b')) \leftrightarrow C$ for some quantifier free C,
5. $S \models \tilde{\forall} C \rightarrow (\exists x(G_1 \wedge \forall y(t^- < y < t^+ \rightarrow G_2))$ if there is a C_1' with $S \models \tilde{\forall}(C_1' \rightarrow G_1)$ and there are C_b' with $S \models \tilde{\forall}(C_b' \rightarrow G_2)$, $b \in I \subseteq \mathbb{Q}$ such that $\exists x(C_1' \wedge \forall y(\{t^- \leqslant^- y \leqslant^+ t^+\} \rightarrow \bigvee_{b \in I} C_b')) \leftrightarrow C$ for some quantifier free C.

The existential quantifier in $\exists x C_1$ can be handled by the Fourier method, the bounded universal quantifiers by the LR-lemma and the procedure EQ introduced in [24] (Fig. 8), the disjunctions can be eliminated relying on the $\bigcup$-lemma (7.9). In order to do this effectively it remains to be shown that it satisfies to consider finite disjunctions. As in the discrete case, we rest upon the notion of tree constraint systems and specializations of the Fourier's algorithm for their satisfiability checking.

In order to prove compactness of bounded universal Horn formulae with respect to bounded universal goals, we show how bounding hyper-planes of the solution sets of the ETCSs can be computed using an extension of the $\rightarrow_{edsc}$ relation.

(QILB)	$C \cup \{c_1^- \leqslant_1^- \bar{x} + \bar{y}, \bar{y} \leqslant_2^+ c_2^+\} \rightarrow_{dqsc}$ $C \cup \{c_1^- \leqslant_1^- \bar{x} + \bar{y}, \bar{y} \leqslant_2^+ c_2^+, c_1^- - c_2^+ \leqslant_1^- \downarrow \leqslant_2^+ \bar{x}\}$
(QIUB)	$C \cup \{\bar{x} + \bar{y} \leqslant_1^+ c_1^+, c_2^- \leqslant_2^- \bar{y}\} \rightarrow_{dqsc}$ $C \cup \{\bar{x} + \bar{y} \leqslant_1^+ c_1^+, c_2^- \leqslant_2^- \bar{y}, \bar{x} \leqslant_1^+ \downarrow \leqslant_2^- c_1^+ - c_2^-\}$
(MLB1)	$C \cup \{c_1 \leqslant_1 \bar{x}, c_2 \leqslant_2 \bar{x}\} \rightarrow_{dqsc} C \cup \{c_1 \leqslant_1 \bar{x}\}$ if $c_1 > c_2$
(MLB2)	$C \cup \{c_1 \leqslant_1 \bar{x}, c_1 \leqslant_2 \bar{x}\} \rightarrow_{dqsc} C \cup \{c_1 \leqslant_1 \downarrow \leqslant_2 \bar{x}\}$
(MUB1)	$C \cup \{\bar{x} \leqslant_1 c_1, \bar{x} \leqslant_2 c_2\} \rightarrow_{dqsc} C \cup \{\bar{x} \leqslant_1 c_1\}$ if $c_1 < c_2$
(MUB2)	$C \cup \{\bar{x} \leqslant_1 c_1, \bar{x} \leqslant_2 c_1\} \rightarrow_{dqsc} C \cup \{\bar{x} \leqslant_1 \downarrow \leqslant_2 c_1\}$

Fig. 13. Satisfiability checking for quasi-tree constraint systems.

Theorem 10.1. *Let C be tree constraint system reduced with respect to $\rightarrow_{dsc}$ and the (DCB) rule. Then the bounds in C for each $\bar{x}$ are precise, i.e.,*

$$c^- \leqslant^- \bar{x} \leqslant^+ c^+ \in C$$

implies that for each c with $c^- \leqslant^- c \leqslant^+ c^+$ there is a solution α of C such that $\bar{\alpha}(\bar{x}) = c$.

Proof. First notice that the precise bounds for each $\bar{x}$ in C can be computed by introducing a new variable z, a new inequality $0 \leqslant -z + \bar{x} \leqslant 0$ into C, and by applying Fourier's method with z being eliminated at the end.

To compute these bounds without destroying the tree constraint property, we can use the simplification rules listed in Fig. 13, which are complete for systems of inequalities constraining (arbitrary) paths of the underlying tree, i.e. which may also constraint suffixes of the paths of the underlying tree (*quasi tree constraint systems*, cf. Appendix B).

It can be shown by induction on the $\rightarrow_{dqsc}$-simplification relation that if

$$C' = \{0 \leqslant -z + \bar{x} \leqslant 0\} \rightarrow_{dqsc} C'' \cup \{c^- \leqslant^- -z + \bar{v} \leqslant^+ c^+\}$$

then $-c^+ \leqslant^+ \bar{x} - \bar{v} \leqslant^- -c^-$ is derivable from C' via the (DCB) rule (Fig. 14) which proves the claim for $\bar{v} = \lambda$. □

Given a set S of extended CLP-goals or extended CLP-formulae, we denote by $KI(S)$ the maximal rational number, which is a factor of all rational numbers occurring in S, i.e., $KI(S) = 1/n$ if n is the least common denominator of all numbers in S. A (extended) tree constraint system C' *extends C on the branch* $\bar{x}$, $C' \rhd_{\bar{x}} C$, if $C' \supseteq C$ and $C' \backslash C$ contains no inequalities for strict prefixes of $\bar{x}$.

$$(DCB)\ C \cup \{c_1^- \preccurlyeq_1^- \bar{x} \preccurlyeq_1^+ c_1^+, c_y^- \preccurlyeq_y^- y \preccurlyeq_y^+ c_y^+\} \rightarrow_{pdsc} \{c_1^- \preccurlyeq_1^- \bar{x} \preccurlyeq_1^+ c_1^+,$$
$$c_y^- \preccurlyeq_y^- y \preccurlyeq_y^+ c_y^+, c_1^- + c_y^- \preccurlyeq_1^- \downarrow \preccurlyeq_y^- \bar{x} + y \preccurlyeq_1^+ \downarrow \preccurlyeq_y^+ c_1^+ + c_y^+\},$$

if for all $c_3^- \preccurlyeq_3^- \bar{x} + y \preccurlyeq_3^+ c_3^+ \in C$, $(c_3^-, \preccurlyeq_3^-) < (c_1^- + c_y^-, \preccurlyeq_1^- \downarrow \preccurlyeq_y^-)$ or for all $c_3^- \preccurlyeq_3^- \bar{x} + y \preccurlyeq_3^+ c_3^+ \in C$, $(c_3^+, \preccurlyeq_3^+) > (c_1^+ - c_y^+, \preccurlyeq_1^+ \downarrow \preccurlyeq_y^+)$.

Fig. 14. Computation of bounding hyper-planes for tree constraint systems.

Lemma 10.2 (Compactness). *Let $S = \bigcup\{\leftarrow C_i \wedge p_i(\bar{x}_i)\}$ be a tree presentation containing only p multiples in its constraint part for some $p \in \mathbb{Q}$, $G = \pi(G', \bar{x}, C)$ an extended CLP-goal, and C a tree constraint systems constraining only variables in $\bar{x}$.*

If $S \models G\theta$, then there exists a finite $S' \subseteq S$ and a tree constraint system C' containing only $KI = KI(S \cup G)$ multiples and with $C' \rhd_{\bar{x}} C$ such that

$$S' \models \tilde{\forall} C' \rightarrow G$$

and $\theta \in [\![C']\!]$.

Proof. We prove compactness by structural induction on G.

The cases $G = C$, $G = A$, and $G = G_1 \wedge G_2$ are straightforward. For the remaining cases, first notice that constraint simplification via $\rightarrow_{dsc}$, $\rightarrow_{edsc}$, $\rightarrow_{dqsc}$, and the (DCB) rule do not perform any divisions operations such that if $C \rightarrow C'$ by some of these relations and C contains only KI multiples, then also C'. Secondly, if $C \rhd_{\bar{x}} C'$ and $C' \vdash_F C''$ via $\rightarrow_{dsc}$ or $\rightarrow_{edsc}$ for some variable not occurring in $\bar{x}$, then $C \rhd_{\bar{x}} C''$.

For G being of the form $\exists x G'$, $S \models \exists x G'\theta$ implies that there is a $b \in \mathbb{Q}$ such that $S \models G'\theta\{x \leftarrow b\}$. Applying inductive hypothesis, there is a TCS $C' \rhd_{\bar{x}} C$ containing only KI multiples and a finite $S' \subseteq S$ with $S' \models \tilde{\forall} C' \rightarrow G'\theta\{x \leftarrow b\}$, $\theta\{x \leftarrow b\} \in [\![C']\!]$. Eliminating x from C' via $\rightarrow_{dsc}$ we obtain a TCS C'' containing only KI multiples, for which $\exists x C' \leftrightarrow C''$, $\theta \in [\![C'']\!]$, and $S' \models \tilde{\forall} C'' \rightarrow \exists x G'$. Since $C' \rhd_{\bar{x}} C$, $C' \vdash_F C''$ by elimination of x via $\rightarrow_{dsc}$, $C'' \rhd_{\bar{x}} C$.

For $G = \forall x(c^- \preccurlyeq^- x \preccurlyeq^+ c^+ \rightarrow G')$, $S \models G$, implies $S \models \pi(G', \bar{x} + x, C)\theta\{x \leftarrow b\}$ for all $b \in \mathbb{Q}$ within $c^- \preccurlyeq^- b \preccurlyeq^+ c^+$. For each of these $\pi(G', \bar{x} + x, C)\theta\{x \leftarrow b\}$ there are, by inductive hypothesis, TCSs $C_b \rhd_{\bar{x}+x} C$, finite $S_b \subseteq S$ such that

$$S_b \models \tilde{\forall} C_b \rightarrow \pi(G', \bar{x} + x, C),$$

$\theta\{x \leftarrow b\} \in [\![C_b]\!]$, and C_b contain only KI multiples. We eliminate all variables from C_b except those in C and x via $\rightarrow_{dsc}$ and obtain $C_b' = C \cup \{c_b^- \preccurlyeq_b^- \bar{x} + x \preccurlyeq_b^+ c_b^+\}$, since $C_b \rhd_{\bar{x}+x} C$. We have

$$S \models \tilde{\forall} \bigvee_{c^- \preccurlyeq^- b \preccurlyeq^+ c^+} C_b' \rightarrow \pi(G', \bar{x} + x, C). \tag{25}$$

We simplify the disjunction

$$\bigvee_{c^- \preccurlyeq^- b \preccurlyeq^+ c^+} C'_b = \bigvee_{c^- \preccurlyeq^- b \preccurlyeq^+ c^+} [C \wedge (c_b^- \preccurlyeq_b^- \bar{x} + x \preccurlyeq_b^+ c_b^+)]$$

to

$$C \wedge \bigvee_{c^- \preccurlyeq^- b \preccurlyeq^+ c+} (c_b^- \preccurlyeq_b^- \bar{x} + x \preccurlyeq_b^+ c_b^+).$$

Suppose that c_b^-, c_b^+ are precise bounds computed with $\rightarrow_{dsc}$ and the (DCB) rule. If $\lfloor \theta(x_i) \rfloor_{KI}$ denotes the greatest *KI* multiple less or equal $\theta(x_i)$ and $\lceil \theta(x_i) \rceil_{KI}$ the smallest *KI* multiple greater or equal $\theta(x_i) \in \mathbb{Q}$, then

$$d^- \preccurlyeq_{\bar{x}}^- \bar{x} \preccurlyeq_{\bar{x}}^+ d^+ \in C \Rightarrow \begin{pmatrix} d^- \leqslant \lfloor \bar{\theta}(\bar{x}) \rfloor_{KI}, \\ \lceil \bar{\theta}(\bar{x}) \rceil_{KI} \leqslant d^+ \end{pmatrix} \tag{26}$$

such that for each solution θ of C and a $\bar{x}$, there is also a solution θ' of C with $\theta'(\bar{x}) = c$ for each c within $\lfloor \theta(\bar{x}) \rfloor_{KI} \preccurlyeq_{\bar{x}}^- c \preccurlyeq_{\bar{x}}^+ \lceil \theta(\bar{x}) \rceil_{KI}$.

On the other hand, it is sufficient to consider c_b^- with $c_b^- \leqslant \lfloor \bar{\theta}(\bar{x}) \rfloor_{KI} + c^-$ and c_b^+ with $\lceil \bar{\theta}(\bar{x}) \rceil_{KI} + c^+ \leqslant c_b^+$ (LR-lemma). Since there are only finitely many *KI* multiples between $\lfloor \bar{\theta}(\bar{x}) \rfloor_{KI} + c^-$ and $\lceil \bar{\theta}(\bar{x}) \rceil_{KI} + c^+$, we can choose $c_{b_i}^-, c_{b_i}^+$, $i = 1, \ldots, n$, such that

$$\underbrace{\bigcup_{i=1}^{n} |_{b_i} c_{b_i}^-, c_{b_i}^+ |_{b_i}}_{I} \supseteq |_{\preccurlyeq^-} \lfloor \bar{\theta}(\bar{x}) \rfloor_{KI} + c^-, \qquad \lceil \bar{\theta}(\bar{x}) \rceil_{KI} + c^+ |_{\preccurlyeq^+}, \tag{27}$$

with $|_{b_i} c_{b_i}^-, c_{b_i}^+ |_{b_i}$ being neighbored or overlapping intervals and I is an interval, i.e., a convex set. $|$ stands here for (or [(respectively, for) or]).

$\theta\{x \leftarrow b\}$ satisfies $C_1 = C \wedge \bigvee_{i=1}^{n} (c_{b_i}^- \preccurlyeq_{b_i}^- \bar{x} + x \preccurlyeq_{b_i}^+ c_{b_i}^+)$ for all b within $c^- \preccurlyeq^- b \preccurlyeq^+ c^+$. We take $\min = \min\{c_{b_1}^-, \ldots, c_{b_n}^-\}$, $\max = \max\{c_{b_1}^+, \ldots, c_{b_n}^+\}$, and rewrite

$$\begin{aligned} C_1 &\leftrightarrow C \wedge \min \preccurlyeq_{\min}^- \bar{x} + x \preccurlyeq_{\max}^+ \max \\ &= C'. \end{aligned}$$

C' is a TCS with $C' \triangleright_{\bar{x}+x} C$ and $\theta \in [\![C']\!]$; C' contains only *KI* multiples and

$$\bigcup_{i=1}^{n} S_{b_i} \models \tilde{\forall} C' \rightarrow \pi(G', \bar{x} + x, C).$$

Applying LR-lemma with $c^- \preccurlyeq^- \downarrow \preccurlyeq^+ c^+$,

$$\begin{gathered} \forall x (c^- \preccurlyeq^- x \preccurlyeq^+ c^+ \rightarrow C') \\ \leftrightarrow \\ EQ(\forall x (c^- \preccurlyeq^- x \preccurlyeq^+ c^+ \rightarrow C')) \end{gathered}$$
$$= C \cup \{\bar{x} \preccurlyeq_{EQ_r} \max - c^+, \min - c^- \preccurlyeq_{EQ_l} \bar{x}\} = C''$$

for suitable $\preccurlyeq_{EQ_r}, \preccurlyeq_{EQ_l} \in \{\leqslant, <\}$. θ is a solution of C'', since per construction (27) $\min \preccurlyeq_{EQ_l} \bar{\theta}(\bar{x}) + c^-$ and $\bar{\theta}(\bar{x}) + c^+ \preccurlyeq_{EQ_r} \max$. In conclusion

$$S' = \bigcup_{i=1}^{n} S_{b_i} \models \tilde{\forall} C'' \rightarrow \forall x(c^- \preccurlyeq^- x \preccurlyeq^+ c^+ \rightarrow \pi(G', \bar{x} + x, C)),$$

$\theta \in [\![C'']\!]$, $C'' \rhd_{\bar{x}} C$, and C'' contains only *KI* multiples.

For $G = \exists x(\pi(G_1, \bar{x}+x, C') \wedge \forall y(c^- + x < y < c^+ \rightarrow \pi(G_2, \bar{x}+x, C)))$, there is a $b \in \mathbb{Q}$ such that

$$S \models \underbrace{\pi(G_1, \bar{x} + x, C') \wedge \forall y(c^- + x < y < c^+ \rightarrow \pi(G_2, \bar{x} + y, C))\theta\{x \leftarrow b\}}_{\pi(G_1, \bar{x}+x, C')\theta\{x \leftarrow b\} \wedge \forall y(c^- + b < y < c^+ \rightarrow \pi(G_2, \bar{x}+y, C)\theta)}.$$

Inductive hypothesis applied to $G_1\theta\{x \leftarrow b\}$ yields the existence of a TCS $C_1 \rhd_{\bar{x}+x} C'$ with $\theta\{x \leftarrow b\} \in [\![C_1]\!]$ containing only *KI* multiples, and of a finite $S_b \subseteq S$ such that $S_b \models \tilde{\forall} C_1 \rightarrow G_1$. We eliminate all variables in C_1 except that in C and x, and obtain

$$\underbrace{C \cup \{c_x^- \preccurlyeq_x^- x \preccurlyeq_x^+ c_x^+, c_{\bar{x}+x}^- \preccurlyeq_{\bar{x}+x}^- \bar{x} + x \preccurlyeq_{\bar{x}+x}^+ c_{\bar{x}+x}^+\}}_{C_1'}.$$

For the second conjunct, we have

$$S \models \pi(G_2, \bar{x} + y, C)\theta\{y \leftarrow b_y\}$$

for all b_y with $c^- + b < b_y < c^+$. Again, by inductive hypothesis there are TCSs $C_{b_y} \rhd_{\bar{x}+y} C$, finite S_{b_y} such that

$$S_{b_y} \models \tilde{\forall} C_{b_y} \rightarrow \pi(G_2, \bar{x} + y, C),$$

$\theta\{y \leftarrow b_y\} \in [\![C_{b_y}]\!]$ and C_{b_y} contains only *KI* multiples. We eliminate all variables in C_{b_y} except that in C and y, and get $C'_{b_y} = C \cup \{c_{b_y}^- \preccurlyeq_{b_y}^- \bar{x} + y \preccurlyeq_{b_y}^+ c_{b_y}^+\}$, since C does not contain inequalities for y. Following the arguments of the preceding case, we conclude that there are $S_{b_{y_i}}$, $i = 1, \ldots, n$, such that

$$\bigcup_{i=1}^{n} S_{b_{y_i}} \models \tilde{\forall} \underbrace{\bigvee_{i=1}^{n} C'_{b_{y_i}}}_{\leftrightarrow C \wedge \bigvee_{i=1}^{n} \{c_{b_{y_i}}^- \preccurlyeq_{b_{y_i}}^- \bar{x}+y \preccurlyeq_{b_{y_i}}^+ c_{b_{y_i}}^+\} = D} \rightarrow \pi(G_2, \bar{x} + y, C)$$

and D is satisfied by $\theta\{x \leftarrow b_y\}$ for all b_y with $c^- + b < b_y < c^+$. Moreover, $D \leftrightarrow C \wedge \{\min \preccurlyeq_{\min}^- \bar{x} + y \preccurlyeq_{\max}^+ \max\} = C_2$ with $\min = \min\{c_{b_{y_1}}^-, \ldots, c_{b_{y_n}}^-\}$ and $\max = \max\{c_{b_{y_1}}^+, \ldots, c_{b_{y_n}}^+\}$. C_2 is a TCS, it contains only *KI* multiples and

$$\forall y(c^- + x < y < c^+ \rightarrow C_2) \;\leftrightarrow\; C \cup \{\min - \bar{x} \leqslant c^- + x, c^+ \leqslant \max - \bar{x}\} = C_2'$$

by LR-lemma. The ETCS $C_1' \cup C_2'$ satisfies

1. $\theta\{x \leftarrow b\} \in [\![C_1' \cup C_2']\!]$.
2. $\bigcup_{i=1}^{n} S_{b_{y_i}} \models \tilde{\forall} C_1' \cup C_2' \rightarrow \forall y(c^- + x < y < c^+ \rightarrow \pi(G_2, \bar{x} + y, C))$, since this is the case for all solutions of $C_2' \subseteq C_1' \cup C_2'$.

3. $S_b \models \tilde{\forall} C_1' \cup C_2' \rightarrow \pi(G_1, \bar{x} + x, C)$.
4. We eliminate x from $C_1' \cup C_2' \vdash_F C_3$ via $\rightarrow_{edsc}$ such that $\exists x(C_1' \cup C_2') \leftrightarrow C_3$. C_3 is a TCS with $C_3 \rhd_{\bar{x}} C$, and $\theta \in [\![C_3]\!]$ due to 1.

2, 3, and 4 imply

$$S_b \cup \bigcup_{i=1}^{n} S_{b_{y_i}} \models \tilde{\forall} C_3 \rightarrow (\exists x(\pi(G_1, \bar{x} + x, C)$$
$$\wedge \forall y(c^- + x < y < c^+ \rightarrow \pi(G_2, \bar{x} + y, C)))),$$

C_3 contains only KI multiples. □

Lemma 10.3. *For each translated bounded universal Horn program P, if $\leftarrow C \wedge p(\bar{x}) \in T_{(P,\mathcal{T})} \uparrow \alpha$, then C is a TCS which contains only $KI(P)$ multiples.*

Proof. We prove the lemma by transfinite induction. The basis with $\alpha = 0$ is trivial. For α being a limit ordinal $T_{(P,\mathcal{T})} \uparrow \alpha = \bigcup_{\beta<\alpha} T_{(P,\mathcal{T})} \uparrow \beta$ allowing to apply the inductive hypothesis directly.

For α being a successor ordinal, $\leftarrow C \wedge p(\bar{x}) \in T_{(P,\mathcal{T})} \uparrow \alpha$ implies the existence of an extended CLP-formulae

$$\pi\left(A, \sum_{i=1}^{n+1} x_i + b\right) \leftarrow \underbrace{\bigcup_{i=1}^{n+1} \{c_i^- \preccurlyeq_i^- x_i \preccurlyeq_i^+ c_i^+\} \wedge \bigwedge_{i=1}^{n} \pi\left(B_i, \sum_{j=1}^{i} x_j, C_i\right)}_{G},$$

where C_i are TCSs constraining only variables from $\sum_{j=1}^{i} x_j$ and $T_{(P,\mathcal{T})} \uparrow (\alpha - 1) = S \models \tilde{\forall} C' \rightarrow G$ for some satisfiable C'.

By definition $S \models \tilde{\forall} C' \rightarrow G$ implies $C' = C_1' \cup \cdots \cup C_n'$ such that

$$S \models \tilde{\forall} C_i' \rightarrow \pi\left(B_i, \sum_{j=1}^{i} x_j, C_i\right).$$

Due to Lemma 10.2 and inductive hypothesis C_i' are TCSs containing only $KI(P)$ multiples and $C_i' \rhd C_i$. Accordingly,

$$C' = \bigcup_{i=1}^{n} C_i',$$
$$C'' = \bigcup_{i=1}^{n+1} \{c_i^- \preccurlyeq_i^- x_i \preccurlyeq_i^+ c_i^+\} \cup C', \quad \text{and}$$
$$C = C'' \cup \left\{x = \sum_{i=1}^{n+1} x_i + b\right\}$$

are TCSs which contain only $KI(P)$ multiples. □

Lemma 10.4. $T_{(P,\mathcal{A}_2)} = [T_{(P,\mathcal{T})}]$.

As a direct consequence of Lemmas 10.4 and 10.3 we get

Lemma 10.5. *Each $T_{(P,\mathscr{A}_2)} \uparrow n$ is finitely presentable by a tree presentation containing only $KI(P)$ multiples.*

Lemma 10.6. *If $\bigcup_{\beta<\alpha} T_{(P,\mathscr{A}_2)} \uparrow \beta \models G(\bar{x})$, then $T_{(P,\mathscr{A}_2)} \uparrow \beta \models G(\bar{x})$ for some $\beta < \alpha$.*

Proof. $S = \bigcup_{\beta<\alpha} T_{(P,\mathscr{T})}\uparrow\beta$ is a tree presentation containing only $KI(P)$ multiples according to Lemma 10.3. By assumption $S \models G(\bar{x})\theta$ for some θ. Due to Lemma 10.2, there is a finite $S' \subseteq S$, a TCS C' with $KI(P)$ multiples and $\theta \in [\![C']\!]$ such that $S' \models \tilde{\forall} C' \rightarrow G(\bar{x})$. Since $S' = \bigcup_{i=1}^{n} \{ \leftarrow C_i \wedge p(\bar{x})\}$ is finite, there is a β_i with $\leftarrow C_i \wedge p_i(\bar{x}) \in T_{(P,\mathscr{T})} \uparrow \beta_i$ for each $i = 1, \dots, n$, by which $S' \subseteq T_{(P,\mathscr{T})} \uparrow \max\{\beta_1, \dots, \beta_n\}$. $\square$

Lemma 10.7. *The following holds:*
1. $T_{(P,\mathscr{T})} \uparrow \omega = lfp(T_{(P,\mathscr{T})})$.
2. $T_{(P,\mathscr{A}_2)} \uparrow \omega = lfp(T_{(P,\mathscr{A}_2)})$.

11. Operational semantics

In this section we present two proof methods for (dense) bounded universal Horn formulae. The first is a direct generalization of the method for the discrete case and is based on a lazy version of the quantifier elimination procedure for bounded universal and existential quantifiers over linear inequalities presented already in Section 8. The second relies on the bounding hyper-planes representation of solution sets of ETCSs already utilized in the proof of the compactness-lemma. It avoids some indeterminisms of the former and admits an elegant integration of constructive negation which will be presented in the next section.

11.1. A direct proof method

The first calculus is defined by the dense time version of MTL-resolution, called *DMTL-resolution*, and which is defined in Fig. 15 in conjunction with the inference rules listed in Fig. 16. We name it EDMTL and denote its derivation relation by $\vdash_{EDMTL}$.

Lemma 11.1. *If $G \vdash_{EDMTL} I$, then I is an ETCS and it contains only $KI(\Pi(P) \cup G)$ multiples.*

Proof. Notice that no of the rules defining EDMTL introduce new constants not already present in their premises, that they keep the ETCS property, and that this is also the case for the procedures EQ, Fig. 18, and $\preccurlyeq$, Fig. 17. $\square$

Lemma 11.2. *If $T_{(P,\mathscr{T})} \uparrow n \models G\theta$, then $G \vdash_{EDMTL} I$ for some satisfiable I with $\theta \in [\![I]\!]$.*

$$\pi(A',\textstyle\sum_{i=1}^{n'+1}x_i + b) \leftarrow \bigcup_{i=1}^{n'+1}\{c_i^- \preccurlyeq_i^- x_i \preccurlyeq_i^+ c_i^+\} \cup C' \wedge \bigwedge_{i=1}^{n'}\pi(B_i', \textstyle\sum_{j=1}^{i} x_j)$$
$$\frac{\leftarrow C \wedge \pi(A,\bar{y} + c) \wedge \bigwedge_{i=0}^{n}\pi(B_i,\bar{y}_{n-i})}{\leftarrow (C \cup C' \cup \{c_1^- \preccurlyeq_1^- \bar{y} + \sum_{i=n'+1}^{2} - x_i + c - b \preccurlyeq_1^+ c_1^+\} \cup \bigcup_{i=2}^{n'+1}\{-c_i^+ \preccurlyeq_i^+ - x_i \preccurlyeq_i^- - c_i^-\} \wedge \bigwedge_{i=1}^{n'}\pi(B_i',\bar{y} + \sum_{j=n+1}^{i+1}(-x_j) + c - b) \wedge \bigwedge_{i=0}^{n}\pi(B_i,\bar{y}_{n-i}))\theta,}$$

where θ is the mgu of A and A', $\bar{y}_{n-i} = \sum_{j=1}^{n-i} y_j$ and $\bar{y} = \bar{y}_n$.

Fig. 15. DMTL-resolution.

Proof. We proof the lemma by induction on n. The basis case, $n = 0$, is trivial. For $n > 0$, we proceed by structural induction on G. The cases G being of the form $C \wedge A$, $G_1 \wedge G_2$, and $\exists x G$ are shown by standard arguments.

For G being of the form $\forall x(c^- \preccurlyeq^- x \preccurlyeq^+ c^+ \rightarrow \pi(G, \bar{x} + x, C))$, $T_{(P,\mathscr{T})} \uparrow n \models G$ implies

$$T_{(P,\mathscr{T})} \uparrow n \models \pi(G, \bar{x} + x, C)\theta\{x \leftarrow b\}$$

for all b within $c^- \preccurlyeq^- b \preccurlyeq^+ c^+$. If $\neg(c^- \preccurlyeq^- \downarrow \preccurlyeq^+ c^+)$ then $(\forall E1)$ is applicable and $c^- \succcurlyeq(\preccurlyeq_1, \preccurlyeq_2)\ c^+$ is true such that $I = \emptyset$ with $\theta = \emptyset$ can be chosen. If $c^- \preccurlyeq^- \downarrow \preccurlyeq^+ c^+$, then applying inductive hypothesis, there are EDMTL-derivations

$$\pi(G, \bar{x} + x, C) \vdash_{EDMTL} I_b \tag{28}$$

with $\theta\{x \leftarrow b\} \in [\![I_b]\!]$ and $[\![I_b]\!]$ contain only $KI(\Pi(P) \cup G)$ multiples (Lemma 11.1). Let consider the tree presentation

$$S = \bigcup\{\leftarrow I_b \wedge p(\bar{x} + x)\}$$

for a new predicate symbol p, for which according to (28)

$$S \models p(\bar{x} + x)\theta\{x \leftarrow b\}$$

for all b within $c^- \preccurlyeq^- b \preccurlyeq^+ c^+$ and consequently

$$S \models \forall x(c^- \preccurlyeq^- x \preccurlyeq^+ c^+ \rightarrow p(\bar{x} + x))\theta.$$

Due to the compactness lemma there is a finite $S' = \bigcup_{i=1}^{n}\{\leftarrow I_{b_i} \wedge p(\bar{x} + x)\} \subseteq S$ with

$$S' \models \forall x(c^- \preccurlyeq^- x \preccurlyeq^+ c^+ \rightarrow p(\bar{x} + x))\theta \tag{29}$$

$$(\forall I)\quad \frac{G[\forall y\,(\{t^- \preccurlyeq^- y \preccurlyeq^+ t^+\} \rightarrow \pi(B,\bar{x}+y))]}{G[\forall y\,(\{t^- \preccurlyeq^- y \preccurlyeq^+ t^+\} \rightarrow \pi(B,\bar{x}+y)\,|\,)]}$$

$$(\exists I)\quad \frac{G[\forall y\,(\{t^- \preccurlyeq^- y \preccurlyeq^+ t^+\} \rightarrow \pi(B,\bar{x}+y)\,|\,\bigwedge_i B_y^i)]}{\begin{array}{l}G[\forall y\,(\{t^- \preccurlyeq^- y \preccurlyeq^+ t^+\} \rightarrow \pi(B,\bar{x}+y)\,|\\ (\pi(B,\bar{x}+y) \wedge \{t^- \preccurlyeq^- y \preccurlyeq^+ t^+\})\theta \wedge \bigwedge_i B_y^i)]\end{array}}$$

for a permutation θ renaming variables in $Var(\pi(B,\bar{x}+y)) \setminus Var(\bar{x})$ into new ones.

$$(\exists\forall C)\quad \frac{\begin{array}{l}\leftarrow G[\forall y\,(\{t^- \preccurlyeq^- y \preccurlyeq^+ t^+\} \rightarrow \pi(B,\bar{x}+y)\,|\\ \{c_1^- \preccurlyeq_1^- y_1 \preccurlyeq_1^+ c_1^+\} \cup I_1 \wedge \bigwedge_i B_y^i)]\end{array}}{\begin{array}{l}\leftarrow G[\forall y\,(\{t^- \preccurlyeq^- y \preccurlyeq^+ t^+\} \rightarrow \pi(B,\bar{x}+y)\,|\\ \bigcup_{\pi=\theta,\sigma}(\{c_1^- \preccurlyeq^- y_1 \preccurlyeq_1^+ c_1^+\} \cup I_1)\pi \cup \{y\theta \leqslant y\sigma\} \wedge \bigwedge_i B_y^i)],\end{array}}$$

for permutations θ and σ renaming variables in $Var(I_1 \cup \{c_1^- \preccurlyeq^- y_1 \preccurlyeq^+ c_1^+\}) \setminus Var(\bar{x})$ into new distinct ones.

$$(\forall C)\quad \frac{\begin{array}{l}\leftarrow G[\forall y\,(\{t^- \preccurlyeq^- y \preccurlyeq^+ t^+\} \rightarrow \pi(B,\bar{x}+y)\,|\\ \bigcup_{i=1,2}(I_i \cup \{y\theta_i \leqslant y\sigma_i\}) \wedge \bigwedge_i B_y^i)]\end{array}}{\begin{array}{l}\leftarrow G[\forall y\,(\{t^- \leqslant y \leqslant t^+\} \rightarrow \pi(B,\bar{x}+y)\,|\,EQ(y\sigma_1, \preccurlyeq_1 y\theta_2, I_1\\ \cup\{y\theta_1 \leqslant y\sigma_1\}) \cup EQ(y\theta_2, \succcurlyeq_2 y\sigma_1, I_2 \cup \{y\theta_2 \leqslant y\sigma_2\})\\ \cup\{y\sigma_1 \leqslant y\sigma_2, y\theta_1 \leqslant y\theta_2\} \wedge \bigwedge_i B_y^i)]\end{array}}$$

for complementary relations $\preccurlyeq_1, \succcurlyeq_2$, i.e., $\preccurlyeq_1 = <$ iff $\succcurlyeq_2 = \geqslant$, $\preccurlyeq_1 = \leqslant$ iff $\succcurlyeq_2 = >$. The function EQ is defined in Fig. 18.

$$(\forall E1)\quad \frac{\leftarrow G[\forall y\,(\{t^- \preccurlyeq^- y \preccurlyeq^+ t^+\} \rightarrow \pi(B,\bar{x}+y)\,|\,\bigwedge_i B_y^i)]}{\leftarrow G[\{t^- \succcurlyeq(\preccurlyeq^-, \preccurlyeq^+) t^+\}]}$$

if $t^- \succcurlyeq(\preccurlyeq^-, \preccurlyeq^+) t^+$ is satisfiable; $\succcurlyeq(\preccurlyeq^-, \preccurlyeq^+)$ is defined in Fig. 17.

$$(\forall E2)\quad \frac{\begin{array}{l}\leftarrow G[\forall y\,(\{t^- \preccurlyeq^- y \preccurlyeq^+ t^+\} \rightarrow \pi(B,\bar{x}+y)\,|\\ I \cup \{y\theta \leqslant y\sigma\} \wedge \bigwedge_i B_y^i)]\end{array}}{\leftarrow G[EQ(y\theta, \succcurlyeq^- t^-, EQ(y\sigma, \preccurlyeq^+ t^+, I \cup \{y\theta \leqslant y\sigma\}))],}$$

if $EQ(y\theta, \succcurlyeq^- t^-, EQ(y\sigma, \preccurlyeq^+ t^+, I \cup \{y\theta \leqslant y\sigma\}))$ is satisfiable.

Fig. 16. Extended DMTL-resolution for proving goals with $\Box_I$, $\mathcal{S}$, and $\mathcal{U}$ operators.

$$\succcurlyeq(\preccurlyeq_1,\preccurlyeq_2)=\begin{cases}\geqslant, & \text{if } \preccurlyeq_1 = < \text{ or } \preccurlyeq_2 = < \\ >, & \text{if } \preccurlyeq_1 = \preccurlyeq_2 = \leqslant\end{cases}$$

Fig. 17. The function $\succcurlyeq(\preccurlyeq_1,\preccurlyeq_2)$.

Function $EQ(y, \bowtie t, C)$

Input: A set of inequalities C, variable y, term t and $\bowtie \in \{\preccurlyeq, \succcurlyeq\}$.

Output: It realizes a delayed elimination of y according to the function EQ given in Fig. 8 which computes either the upper — $\bowtie = \succcurlyeq$ — or the lower — $\bowtie = \preccurlyeq$ — part of the formula determined by EQ, in case $\forall y(t^- \preccurlyeq y \preccurlyeq t^+ \rightarrow \ldots)$ has a satisfiable $t^- \preccurlyeq y \preccurlyeq t^+$, i.e. the range specified is not empty.

begin

Arrange C in a form suitable for elimination of y, i.e., into

$$C_1 = \{l_i \preccurlyeq_i^l y \quad i = 1,\ldots,p$$
$$y \preccurlyeq_j^r r_j \quad j = 1,\ldots,q$$
$$d_i \preccurlyeq_i^d 0 \quad i = 1,\ldots,s\}.$$

$$C_2 = \{l_i \preccurlyeq_i^l \downarrow \preccurlyeq_j^r r_j \mid i = 1,\ldots,p; j = 1,\ldots,q\} \cup \{d_i \preccurlyeq_i^d 0 \mid i = 1,\ldots,s\}.$$

if $\bowtie = \succcurlyeq$ **then** $C_l =$ **if** $\bowtie = >$ **then** $\bigcup_{i=1}^p \{l_i \leqslant t\}$
else $\bigcup_{i=1}^p \{l_i \preccurlyeq_i^l t\}$
else $C_l = \emptyset$
if $\bowtie = \preccurlyeq$ **then** $C_r =$ **if** $\bowtie = <$ **then** $\bigcup_{j=1}^q \{t \leqslant r_j\}$
else $\bigcup_{j=1}^q \{t \preccurlyeq_j^+ r_j\}$
else $C_r = \emptyset$
return$(C_2 \cup C_l \cup C_r)$

end

Fig. 18. Delayed elimination of bounded quantifiers with variable bounds.

and as a consequence an EDMTL-derivation of the form

$$\forall x(c^- \preccurlyeq^- x \preccurlyeq^+ c^+ \rightarrow \pi(G, \bar{x} + x, C))$$

$$\vdash_{\forall I} \forall x(c^- \preccurlyeq^- x \preccurlyeq^+ c^+ \rightarrow \pi(G, \bar{x} + x, C) \mid)$$

$$\vdash_{\exists I} \forall x(c^- \preccurlyeq^- x \preccurlyeq^+ c^+ \rightarrow \pi(G, \bar{x} + x, C) \mid \bigwedge_{i=1}^{n}(\{c^- \preccurlyeq^- y \preccurlyeq^+ c^+\} \wedge \pi(G, \bar{x} + x, C))\theta_{b_i})$$

$$\vdash_{DMTL} \forall x(c^- \preccurlyeq^- x \preccurlyeq^+ c^+ \rightarrow \pi(G, \bar{x}+x, C) \mid$$

$$\bigwedge_{i=1}^{n} I'_{b_i} \cup \{c^-_{b_i} \preccurlyeq^-_{b_i} \bar{x} + x\theta_{b_i} \preccurlyeq^+_{b_i} c^+_{b_i}\})$$

$$\vdash_{\exists\forall C} \forall x(c^- \preccurlyeq^- x \preccurlyeq^+ c^+ \rightarrow \pi(G, \bar{x}+x, C) \mid$$

$$\bigwedge_{i=1}^{n} \underbrace{I'_{b_i} \cup \{c^-_{b_i} \preccurlyeq^-_{b_i} \bar{x} + x\theta_{b_i} \preccurlyeq^+_{b_i} c^+_{b_i}\} \cup \{x\theta_{b_i} \leqslant x\sigma_{b_i}\}}_{I''_{b_i}})$$

$$\vdash_{\forall C} \forall x(c^- \preccurlyeq^- x \preccurlyeq^+ c^+ \rightarrow \pi(G, \bar{x}+x, C) \mid$$

$$\left.\begin{aligned} &\bigwedge_{i=1}^{n} EQ(x\sigma_{b_i}, \preccurlyeq_i x\theta_{b_{i+1}}, I''_{b_i} \\ &\quad \cup EQ(x\theta_{b_{i+1}}, \succcurlyeq_{i+1} x\sigma_{b_i}, I''_{b_{i+1}}) \\ &\quad \cup \{x\sigma_{b_i} \leqslant x\sigma_{b_{i+1}}, x\theta_{b_i} \leqslant x\theta_{b_{i+1}}\}) \end{aligned}\right\} I$$

for suitable $\preccurlyeq_i$ and $\succcurlyeq_{i+1}$ (due to (29))

$$\vdash_{\forall E2} \underbrace{EQ(x\theta_{b_1}, \succcurlyeq^- c^-, EQ(x\theta_{b_n}, \preccurlyeq^+ c^+, I))}_{I_1}$$

We have $\theta \in [\![I_1]\!]$ due to (29).

If G is of the form $\exists x(\pi(G_1, \bar{x}+x, C') \wedge \forall y(c^- + x < y < c^+ \rightarrow \pi(G_2, \bar{x}+y, C)))$, then

$$\begin{aligned} T_{(P,\mathscr{T})} \uparrow n \models\ & \exists x(\pi(G_1, \bar{x}+x, C') \wedge \\ & \forall y(c^- + x < y < c^+ \rightarrow \pi(G_2, \bar{x}+y, C)))\theta \text{ iff} \\ T_{(P,\mathscr{T})} \uparrow n \models\ & \pi(G_1, \bar{x}+x, C')\theta\{x \leftarrow b\} \wedge \\ & \forall y(c^- + b < y < c^+ \rightarrow \pi(G_2, \bar{x}+y, C)\theta\{x \leftarrow b\}) \end{aligned}$$

for some b. By inductive hypothesis applied to both parts of the conjunction there are derivations

$$\pi(G_1, \bar{x}+x, C') \vdash_{EDMTL} I_1, \tag{30}$$

$$\forall y(c^- + b < y < c^+ \rightarrow \pi(G_2, \bar{x}+y, C)) \vdash_{EDMTL} I_2 \tag{31}$$

with $\theta\{x \leftarrow b\} \in [\![I_1]\!]$ and $\theta \in [\![I_2]\!]$. Since (31) is of the form

$$\begin{aligned} &\forall y(c^- + b < y < c^+ \rightarrow \pi(G_2, \bar{x}+y, C)) \vdash_{EDMTL} \\ &\forall y(c^- + b < y < c^+ \rightarrow \pi(G_2, \bar{x}+y, C) \mid I \cup \{y\theta \leqslant y\sigma\}) \vdash_{E2} \\ &EQ(y\theta, > c^- + b, EQ(y\sigma, < c^+, I \cup \{y\theta \leqslant y\sigma\})) = I_2 \end{aligned}$$

there is a derivation

$$\begin{aligned}
&\exists x(\pi(G_1,\bar{x}+x,C')\wedge \\
&\forall y(c^- + x < y < c^+ \rightarrow \pi(G_2,\bar{x}+y,C))) \\
\vdash_{EDMTL}\ & I_1 \wedge \forall y(c^- + x < y < c^+ \rightarrow \pi(G_2,\bar{x}+y,C)) \\
\vdash_{EDMTL}\ & I_1 \wedge \forall y(c^- + x < y < c^+ \rightarrow \pi(G_2,\bar{x}+y,C) \mid \\
& I' \cup \{y\theta \leqslant y\sigma\})
\end{aligned}$$

with $I'\{x \leftarrow b\} = I$ which can be continued by

$$\vdash_{E2} I_1 \wedge \underbrace{EQ(y\theta, > c^- + x, EQ(y\sigma, < c^+, I' \cup \{y\theta \leqslant y\sigma\}))}_{I_2'}$$

Since $I_2'\{x \leftarrow b\} \leftrightarrow I_2$, I_2' is satisfiable and $\theta\{x \leftarrow b\} \in [\![I_2']\!]$, such that $\theta\{x \leftarrow b\} \in [\![I_1 \cup I_2]\!]$ implying the satisfiability of $I_1 \cup I_2$. $\square$

Theorem 11.3. *Let P denote a set of (dense) bounded universal Horn formulae and G a (dense) bounded universal goal.*

(Soundness) *If $\Pi(G) \vdash_{EDMTL} I$, then $\Pi(P) \models \tilde{\forall} I \rightarrow \Pi(G)$.*

(Completeness) *If $\Pi(P) \models \Pi(G)$, then $\Pi(G) \vdash_{EDMTL} I$ for some satisfiable I.*

Proof. Soundness follows similarly to the discrete case. Completeness follows directly from Lemma 11.2, since $\Pi(P) \models \Pi(G)$ iff $lm(\Pi(P), \mathscr{A}_2)) = lfp(T_{(\Pi(P),\mathscr{A}_2)} = T_{(\Pi(P),\mathscr{A}_2)} \uparrow \omega \models \Pi(G)$. $\square$

Theorem 11.3 in conjunction with Corollary 5.3 implies soundness and completeness of the *EDMTL* calculus. The complexity result for the discrete case holds also in the dense case.

Theorem 11.4. *Let P denote an extended CLP-program and $C \wedge G$ an extended CLP-goal.*

If $C \wedge G \vdash_{EDMTL} C' \wedge G'$ with a $\rightarrow_{edsc}$-normalized C, then the satisfiability of C' is decidable in linear time via $\rightarrow_{edsc}$.

11.2. Optimized proof method

The optimized proof method for (dense) bounded universal Horn formulae is based on Theorem 10.1 allowing to compute bounding hyper-planes for solution sets of ETCSs efficiently. It can be also used in a simplified form for the discrete case and be seen as an alternative to the EMTL-calculus presented in [24]. The motivation for its development arised during implementation efforts on a temporal logic programming system – LIMETTE – relying upon bounded universal modality Horn formulae [26, 97] which is based on a preliminary version of this calculus for discrete time.

The optimized calculus, subsequently called ODMTL, delays the elimination of bounded quantifiers and gives preference to applications of $\bigcup$-lemma which can be performed on the existential level due to Theorem 10.1. The method itself is presented in Figs. 19 and 20.

Lemma 11.5. *If $G \vdash_{ODMTL} I$, then I is a TCS which contains only $KI(P \cup G)$ multiples.*

Theorem 11.6. *Let P denote a set of (dense) bounded universal Horn formulae and G a (dense) bounded universal goal.*

(Soundness) *If $\Pi(G) \vdash_{ODMTL} I$ from $\Pi(P)$ for some satisfiable I, then $\Pi(P) \models I \to \Pi(G)$ and $P \models G$.*

(Completeness) *If $P \models G$, then $\Pi(P) \vdash_{ODMTL} I$ from $\Pi(P)$ for some satisfiable I.*

Proof. (Soundness) We show, if

$$\Pi(G) \vdash_{ODMTL} I \tag{32}$$

with a satisfiable I, then $\Pi(P) \models I \to \Pi(G)$, which implies $P \models G$ due to the soundness of the translation. The proof goes on by induction on the number n of $(\forall I)$-applications.

For $n = 0$, (32) is a pure $\vdash_{DMTL}$-derivation and soundness follows by soundness of $\vdash_{DMTL} \subseteq \vdash_{(\Pi(P),\mathscr{A}_2)}$, i.e., $\Pi(P) \models I \to \Pi(G)$.

If $n > 0$, then (32) is of the form[12]

$$\begin{aligned}
G = \Pi(G) \;&\vdash_{DMTL}\; G_1[\forall y(t^- \preccurlyeq_t^- y \preccurlyeq_t^+ t^+ \to B(\bar{x}+y))] \\
&\vdash_{(\forall I)}\; G_1[\forall y(t^- \preccurlyeq_t^- y \preccurlyeq_t^+ t^+ \to B(\bar{x}+y)| : \emptyset)] \\
&\vdash_{(\exists I)}\; G_1[\forall y(t^- \preccurlyeq_t^- y \preccurlyeq_t^+ t^+ \to B(\bar{x}+y)|(\{t^- \preccurlyeq_t^- y \preccurlyeq_t^+ t^+\} \\
&\qquad \wedge B(\bar{x}+y))\theta_i : \emptyset] \\
&\quad \vdots \\
&\vdash^*_{(\exists C)}\; G_2 = G[\forall y(t^- \preccurlyeq_t^- y \preccurlyeq_t^+ t^+ \to B(\bar{x}+y)| \\
&\qquad \textstyle\bigwedge B_i : P] \\
&\vdash_{ODMTL} \cdots \\
&\vdash_{DMTL}\; I
\end{aligned} \tag{33}$$

and $G_1[\forall y(t^- \preccurlyeq_t^- y \preccurlyeq_t^+ t^+ \to B(\bar{x}+y)| : \emptyset)] \vdash_{ODMTL} I$ contains at most $n-1$ applications of the $(\forall I)$ rule.

For each $|_{\preccurlyeq^-} c^-, c^+|_{\preccurlyeq^+} \in P$ there are derivations

$$\begin{aligned}
&G_1[\forall y(\{t^- \preccurlyeq_t^- y \preccurlyeq_t^+ t^+\} \to B(\bar{x}+y) \,|\, (\{t^- \preccurlyeq_t^- y \preccurlyeq_t^+ t^+\} \wedge B(\bar{x}+y))\theta : P_\theta] \\
&\vdash_{ODMTL} \\
&G_1[\forall y(\{t^- \preccurlyeq_t^- y \preccurlyeq_t^+ t^+\} \to B(\bar{x}+y) \,|\, \{c^- \preccurlyeq^- \bar{x}+y \preccurlyeq^+ c^+\}\theta : P_\theta)],
\end{aligned} \tag{34}$$

[12] If $G \vdash \ldots \vdash_{(\forall E)} G_2$, soundness follows directly by inductive hypothesis.

$(\forall I)\quad \dfrac{G[\forall y(t^- \leqslant^- y \leqslant^+ t^+ \to B(\bar{x}+y))]}{G[\forall y(t^- \leqslant^- y \leqslant^+ t^+ \to B(\bar{x}+y) \,|\, : \emptyset)]}$

$(\exists I)\quad \dfrac{G[\forall y(t^- \leqslant^- y \leqslant^+ t^+ \to B(\bar{x}+y) \,|\, \bigwedge B_i : P)]}{G[\forall y(t^- \leqslant^- y \leqslant^+ t^+ \to B(\bar{x}+y) | (\{t^- \leqslant^- y \leqslant^+ t^+\} \wedge B(\bar{x}+y))\theta \wedge \bigwedge B_i : P)]}$

for a permutation θ on variables in $Var(B(\bar{x}+y)) \setminus Var(\bar{x})$.

$(\exists C)\quad \dfrac{G[\forall y(t^- \leqslant^- y \leqslant^+ t^+ \to B(\bar{x}+y)) \,|\, \{t^- \leqslant^- y\theta \leqslant^+ t^+, c^- \leqslant^- \bar{x} + y\theta \leqslant^+ c^+\} \wedge \bigwedge B_i : P]}{G[\forall y(t^- \leqslant^- y \leqslant^+ t^+ \to B(\bar{x}+y) \,|\, \bigwedge B_i : \{|_{\leqslant^-} c^-, c^+|_{\leqslant^+}\} \cup P)]}$

if c^-, c^+ are precise bounds of $\bar{x} + y\theta$ computed with $\to_{pdsc}$.

$(\exists\forall C)\quad \dfrac{G[\forall y(c^- \leqslant^- y \leqslant^+ c^+ \to B(\bar{x}+y) \,|\, \bigwedge B_i : P)]}{G[\forall y(c^- \leqslant^- y \leqslant^+ c^+ \to B(\bar{x}+y) \,|\, EQ(\forall y(c^- \leqslant^- y \leqslant^+ c^+ \to \{b^- \leqslant_b^- \bar{x} + y \leqslant_b^+ b^+\})) : P)]}$

if $c^-, c^+ \in \mathbb{Q}$, $|_{\leqslant_b^-} b^-, b^+|_{\leqslant_b^+} \in \bigcup P$ and $EQ(\forall y(c^- \leqslant^- y \leqslant^+ c^+ \to \{b^- \leqslant_b^- \bar{x} + y \leqslant_b^+ b^+\}))$ is satisfiable, where $\bigcup P$ denotes the set of all intervals and their unions in P.

$(\exists\forall \mathscr{S} C)\quad \dfrac{G[\{y<0, c^- \leqslant^- \bar{x} + y \leqslant^+ c^+\} \dot{\wedge} \forall y'(y < y' < 0 \to B(\bar{x}+y') \,|\, \bigwedge B_i : P)]}{G[\max(c^-, b^-) < \bar{x} \leqslant b^+]}$

if c^- and c^+ are precise bounds computed with $\to_{pdsc}$ and $|_{\leqslant_b^-} b^-, b^+|_{\leqslant_b^+} \in \bigcup P$, and $|_{\leqslant^-} c^-, c^+|_{\leqslant^+} \cap |_{\leqslant_b^-} b^-, b^+|_{\leqslant_b^+} \neq \emptyset$ or $c^+ = b^-$ and $\leqslant^+ = \leqslant$.

$(\exists\forall \mathscr{U} C)\quad \dfrac{G[\{0<y, c^- \leqslant^- \bar{x} + y \leqslant^+ c^+\} \dot{\wedge} \forall y'(0 < y' < y \to B(\bar{x}+y') \,|\, \bigwedge B_i : P)]}{G[\{b^- \leqslant \bar{x} < \min(b^+, c^+)\}]}$

if c^-, c^+ are precise bounds computed with $\to_{pdsc}$ and $|_{\leqslant_b^-} b^-, b^+|_{\leqslant_b^+} \in \bigcup P$, and $|_{\leqslant_b^-} b^-, b^+|_{\leqslant_b^+} \cap |_{\leqslant^-} c^-, c^+|_{\leqslant^+} \neq \emptyset$ or $c^- = b^+$ and $\leqslant^- = \leqslant$.

Fig. 19. Optimized proof method for bounded universal Horn formulae, dense version.

whereby c^-, c^+ are precise bounds computed with $\to_{pdsc}$. Let C denote the inequalities from G_1. Then from (34) follows by inductive hypothesis

$$\Pi(P) \models \tilde{\forall}(C \cup \{c_i^- \leqslant_i^- \bar{x} + y \leqslant_i^+ c_i^+\} \to B(\bar{x}+y)) \tag{35}$$

$$(\forall E)\ \frac{\leftarrow G[\forall y\,(\{t^- \preccurlyeq^- y \preccurlyeq^+ t^+\} \rightarrow \pi(B, \bar{x}+y) \mid \bigwedge_i B_y^i)]}{\leftarrow G[\{t^- \succcurlyeq(\preccurlyeq^-, \preccurlyeq^+) t^+\}]}$$

if $t^- \succcurlyeq(\preccurlyeq^-, \preccurlyeq^+) t^+$ is satisfiable; $\succcurlyeq(\preccurlyeq^-, \preccurlyeq^+)$ is defined in Fig. 17.

Fig. 20. Optimized proof method for bounded universal Horn formulae, proving of bounded universal quantified formulae with empty ranges, dense version.

$$(not\ I)\ \frac{G[not\ B(\bar{x})]}{G[not\ B(\bar{x}) \mid : \emptyset]}$$

$$(not\ \exists I)\ \frac{G[not\ B(\bar{x}) \mid \bigwedge B_i : P]}{G[not\ B(\bar{x}) \mid B(\bar{x})\theta \wedge \bigwedge B_i : P]}$$

for a permutation θ on the time variables in $Var(B(\bar{x}))$.

$$(not\ \exists C)\ \frac{G[not\ B(\bar{x}) \mid \{c^- \preccurlyeq^- x\theta \preccurlyeq^+ c^+\} \wedge \bigwedge B_i : P]}{G[not\ B(\bar{x}) \mid \bigwedge B_i : \{|_{\preccurlyeq^-} c^-, c^+|_{\preccurlyeq^+}\} \cup P]}$$

if c^-, c^+ are precise bounds for $\bar{x}\theta$ computed via $\rightarrow_{pdsc}$.

$$(not\ C)\ \frac{G[not\ B(\bar{x}) \mid \bigwedge B_i : P]}{G[\{c^- \preccurlyeq^- \bar{x} \preccurlyeq^+ c^+\}]}$$

if P contains intervals $|_{\preccurlyeq_b^-} b^-, b^+|_{\preccurlyeq_b^+}$ for all successful derivations of $B(\bar{x})\theta_b$ with $\rightarrow_{pdsc}$-normalized answer constraints $\{b^- \preccurlyeq_b^- \bar{x}\theta_b \preccurlyeq_b^+ b^+\}$ and $|_{\preccurlyeq^-} c^-, c^+|_{\preccurlyeq^+} \in \overline{\bigcup P}$; $\overline{\bigcup P}$ denotes the complement of $\bigcup P$ with respect to $\mathbb{Q}$.

Fig. 21. Inference rules for *negation as failure*.

for all $|_{\preccurlyeq_i^-} c_i^-, c_i^+|_{\preccurlyeq_i^+} \in P$, and

$$\Pi(P) \models \tilde{\forall}\left(C \wedge \bigvee_{i=1}^{n} \{c_i^- \preccurlyeq_i^- \bar{x} + y \preccurlyeq_i^+ c_i^+\} \rightarrow B(\bar{x}+y)\right). \tag{36}$$

Since c_i^-, c_i^+ are precise bounds computed via $\rightarrow_{pdsc}$, (36) implies

$$\Pi(P) \models \tilde{\forall}(C \wedge \{b^- \preccurlyeq^- \bar{x} + y \preccurlyeq^+ b^+\} \rightarrow B(\bar{x}+y)) \tag{37}$$

for all $|_{\preccurlyeq^-} b^-, b^+|_{\preccurlyeq^+} \in \bigcup P$.

1. If $t^-, t^+ \in \mathbb{Q}$, then according to the LR-lemma

$$\begin{array}{lcl} \forall y(\{t^- \preccurlyeq_t^- y \preccurlyeq_t^+ t^+\} & \to & C \cup \{b^- \preccurlyeq^- \bar{x} + y \preccurlyeq^+ b^+, t^- \preccurlyeq_t^- y \preccurlyeq_t^+ t^+\}) \\ & \leftrightarrow & \\ EQ(\forall y(\{t^- \preccurlyeq_t^- y \preccurlyeq_t^+ t^+\} & \to & C \cup \{b^- \preccurlyeq^- \bar{x} + y \preccurlyeq^+ b^+, t^- \preccurlyeq_t^- y \preccurlyeq_t^+ t^+\})) \\ & = & \\ EQ(\forall y(\{t^- \preccurlyeq_t^- y \preccurlyeq_t^+ t^+\} & \to & \{b^- \preccurlyeq^- \bar{x} + y \preccurlyeq^+ b^+\})) \cup C \end{array}$$

since y occurs only in $b^- \preccurlyeq^- \bar{x} + y \preccurlyeq^+ b^+$ and $t^- \preccurlyeq_t^- y \preccurlyeq_t^+ t^+$, (33) is continued by $(\exists\forall C)$. Then

$$C \cup \underbrace{EQ(\forall y(\{t^- \preccurlyeq_t^- y \preccurlyeq_t^+ t^+\} \to \{b^- \preccurlyeq^- \bar{x} + y \preccurlyeq^+ b^+\}))}_{I_1}$$

$$\to (\forall y(\{t^- \preccurlyeq_t^- y \preccurlyeq_t^+ t^+\} \to C \cup \{b^- \preccurlyeq^- \bar{x} + y \preccurlyeq^+ b^+, t^- \preccurlyeq_t^- y \preccurlyeq_t^+ t^+\}))$$

and due to (37) also

$$\Pi(P) \models \tilde{\forall} I_1 \cup C \to \forall y(t^- \preccurlyeq_t^- y \preccurlyeq_t^+ t^+ \to B(\bar{x} + y)).$$

We have

$$\Pi(P) \models G_1[\forall y(t^- \preccurlyeq_t^- y \preccurlyeq_t^+ t^+ \to B(\bar{x} + y))] \leftarrow G_1[I_1 \cup C],$$

and due to soundness of $\vdash_{DMTL}$-derivations

$$\Pi(P) \models G \leftarrow G_1[\forall y(t^- \preccurlyeq_t^- y \preccurlyeq_t^+ t^+ \to B(\bar{x} + y))]$$

and

$$\Pi(P) \models G_1[I_1 \cup C] \leftarrow I,$$

implying

$$\Pi(P) \models \tilde{\forall} I \to G.$$

2. If $t^- = y$ for some variable y, $y < 0 \in C$, $\preccurlyeq_t^- = <$, $\preccurlyeq_t^+ = <$, then G_1 has the form

$$\begin{array}{ll} G_1 = & G_1[\exists y(\{y<0, y \in J\} \wedge A(\bar{x} + y) \wedge \forall y'(y < y' < 0 \to B(\bar{x} + y')))] \\ \vdash_{(\forall I)} & G_1[\exists y(\{y<0, y \in J\} \wedge A(\bar{x} + y) \wedge \forall y'(y < y' < 0 \to B(\bar{x} + y')| : \emptyset))] \\ \vdash_{ODMTL} & G_2[\{y<0,\ c^- \preccurlyeq^- \bar{x} + y \preccurlyeq^+ c^+\} \wedge \forall y' \left(y < y' < 0 \to B(\bar{x} + y)|\bigwedge B_i : P\right)] \end{array}$$

by at most $(n-1)$ applications of the $(\forall I)$ rule.

Let C denote the inequalities in G_2 except that in $\{y<0,\ c^- \leqslant^- \bar{x}+y \leqslant^+ c^+\}$ and let assume c^-, c^+ are precise, then

$$\Pi(P) \models \tilde{\forall} C \cup \{y<0, c^- \leqslant^- \bar{x}+y \leqslant^+ c^+\} \rightarrow A(\bar{x}+y) \tag{38}$$

and due to (37) also

$$\Pi(P) \models \tilde{\forall} C \cup \{b^- \leqslant_b^- \bar{x}+y' \leqslant_b^+ b^+\} \rightarrow B(\bar{x}+y') \tag{39}$$

for $|_{\leqslant_b^-} b^-, b^+|_{\leqslant_b^+} \in \bigcup P$.

For each solution α of C with $\max(c^-, b^-) < \bar{\alpha}(\bar{x}) \leqslant b^+$: If

$$|_{\leqslant^-} c^-, c^+|_{\leqslant^+} \cap |_{\leqslant_b^-} b^-, b^+|_{\leqslant_b^+} \neq \emptyset$$

then there is a c with $b^- \leqslant_b^- c \leqslant^+ c^+$ and if $c^+ = b^-$ and $\leqslant^+ = \leqslant$ then there is a $c = c^+ = b^-$ with $c^- \leqslant^- c \leqslant c^+$.

In both cases, we choose $\alpha(y) = c - \bar{\alpha}(\bar{x})$, for which

$$c^- \leqslant^- \underbrace{\bar{\alpha}(\bar{x}+y)}_{\substack{= \bar{\alpha}(\bar{x}) + \alpha(y) \\ = \bar{\alpha}(\bar{x}) + c - \bar{\alpha}(\bar{x}) = c}} \leqslant^+ c^+, \tag{40}$$

and for all $\alpha(y')$ with $\alpha(y) < \alpha(y') < 0$,

$$b^- \leqslant_b^- \bar{\alpha}(\bar{x}) + \alpha(y') \leqslant_b^+ b^+, \tag{41}$$

since $b^- \leqslant c = \bar{\alpha}(\bar{x}) + \underbrace{c - \bar{\alpha}(\bar{x})}_{\alpha(y)} < \bar{\alpha}(\bar{x}) + \alpha(y') \leqslant_b^+ b^+$ due to $\alpha(y') < 0$ and $\bar{\alpha}(\bar{x}) \leqslant b^+$.

Using (38) and (39) we obtain

$$\begin{aligned} \Pi(P) \models \tilde{\forall}(C \cup \{\max(c^-, b^-) < \bar{x} \leqslant b^+\} &\rightarrow \exists y(\{y<0\} \wedge A(\bar{x}+y) \\ &\wedge \forall y'(y<y'<0 \rightarrow B(\bar{x}+y')))), \end{aligned}$$

and thereby also

$$\begin{aligned} \Pi(P) \models G_1[\exists y(\{y<0\} \wedge A(\bar{x}+y) \wedge \forall y'(y<y'<0 \rightarrow B(\bar{x}+y')))] \\ \leftarrow G_1[\{\max(c^-, b^-) < \bar{x} \leqslant b^+\}]. \end{aligned}$$

Soundness of $\vdash_{DMTL}$ and inductive hypothesis implies

$$\Pi(P) \models G \leftarrow G_1[\exists y(\{y<0\} \wedge A(\bar{x}+y) \wedge \forall y'(y<y'<0 \rightarrow B(\bar{x}+y)))],$$

$$\Pi(P) \models G_1[\{\max(c^-, b^-) < \bar{x} \leqslant b^+\}] \leftarrow I,$$

and transitivity of $\leftarrow$ also $\Pi(P) \models G \leftarrow I$.

(*Completeness*). We show

$$T_{(P,\mathscr{T})} \uparrow n \models G\theta \;\Rightarrow\; G \vdash_{ODMTL} I$$

with a satisfiable I and $\theta \in [\![I]\!]$ by induction on n, which shows completeness. The proof has the same structure as that of Lemma 11.2. It differs only for goals obtained by translation of $\Box_i G$, $i \in \mathbb{Q}$, and $A\ \mathcal{S}_J\ B$ (respectively, $A\ \mathcal{U}_J\ B$).

For G being of the form $\forall y(c^- \preccurlyeq^- x \preccurlyeq^+ c^+ \rightarrow \pi(G, \bar{x} + x, C))$, $T_{(P,\mathcal{T})} \uparrow n \models G\theta$ iff $T_{(P,\mathcal{T})} \uparrow n \models \pi(G, \bar{x} + x, C)\theta\{x \leftarrow b\}$ for all b within $c^- \preccurlyeq^- b \preccurlyeq^+ c^+$. If $\neg(c^- \preccurlyeq^- \downarrow \preccurlyeq^+ c^+)$ then $(\forall E)$ is applicable and $c^- \succcurlyeq(\preccurlyeq^-, \preccurlyeq^+)\, c^+$ is true such that $I = \emptyset$ with $\theta = \emptyset$ can be chosen. If $c^- \preccurlyeq^- \downarrow \preccurlyeq^+ c^+$, then according to the inductive hypothesis there are $\vdash_{ODMTL}$-derivations

$$\pi(G, \bar{x} + x, C) \vdash_{ODMTL} I_b \tag{42}$$

with $\theta\{x \leftarrow b\} \in [\![I_b]\!]$, and I_b being TCSs, Lemma 11.5, containing only $KI(\Pi(P) \cup G)$ multiples.

Let

$$S = \bigcup_{c^- \preccurlyeq^- b \preccurlyeq^+ c^+} \{\leftarrow I_b \wedge p(\bar{x} + x)\},$$

be a tree presentation, for which due to (42)

$$S \models \forall y(c^- \preccurlyeq^- y \preccurlyeq^+ c^+ \rightarrow p(\bar{x} + x))\theta.$$

Applying the compactness-lemma there is a finite $S' = \bigcup_{i=1}^{n} \{\leftarrow I_{b_i} \wedge p(\bar{x} + x)\} \subseteq S$, for which $S' \models \forall y(c^- \preccurlyeq^- y \preccurlyeq^+ c^+ \rightarrow p(\bar{x} + x))\theta$ implying

$$\forall y \left(c^- \preccurlyeq^- y \preccurlyeq^+ c^+ \rightarrow \bigvee_{i=1}^{n} I_{b_i} \right) \theta. \tag{43}$$

Each I_{b_i} can be rewritten into

$$I \cup \{c_i^- \preccurlyeq_i^- \bar{x} + y \preccurlyeq_i^+ c_i^+,\ c^- \preccurlyeq^- y \preccurlyeq^+ c^+\}$$

with $y \notin Var(I)$ and c_i^-, c_i^+ being precise. Since

$$I_1 = \bigvee_{i=1}^{n} I_{b_i} \leftrightarrow I \cup \{c^- \preccurlyeq^- y \preccurlyeq^+ c^+\} \wedge \bigvee_{i=1}^{n} \{c_i^- \preccurlyeq_i^- \bar{x} + y \preccurlyeq_i^+ c_i^+\},$$

(43) and due to LR-lemma

$$\bigcup |_{\preccurlyeq_i^-} c_i^-, c_i^+ |_{\preccurlyeq_i^+} \supseteq |_{\preccurlyeq^-} \theta(\bar{x}) + c^-,\ \theta(\bar{x}) + c^+ |_{\preccurlyeq^+}. \tag{44}$$

Accordingly, there is a derivation

$$
\begin{array}{ll}
G = & \forall x(c^- \preccurlyeq^- x \preccurlyeq^+ c^+ \rightarrow \pi(G, \bar{x}+x, C)) \\
\vdash_{\forall I} & \forall x(c^- \preccurlyeq^- x \preccurlyeq^+ c^+ \rightarrow \pi(G, \bar{x}+x, C) : \emptyset) \\
\vdash_{\exists I} & \forall x\left(c^- \preccurlyeq^- x \preccurlyeq^+ c^+ \rightarrow \pi(G, \bar{x}+x, C) \middle| \right. \\
 & \left. \qquad \bigwedge_{i=1}^{n} (\{c^- \preccurlyeq^- x \preccurlyeq^+ c^+\} \wedge \pi(G, \bar{x}+x, C))\theta_{b_i} : \emptyset \right) \\
\vdash_{ODMTL} & \forall x\left(c^- \preccurlyeq^- x \preccurlyeq^+ c^+ \rightarrow \pi(G, \bar{x}+x, C) \middle| \bigwedge_{i=1}^{n} I_{b_i}\theta_{b_i} : \emptyset \right) \\
\vdash_{\exists C} & \forall x\left(c^- \preccurlyeq^- x \preccurlyeq^+ c^+ \rightarrow \pi(G, \bar{x}+x, C) \middle| : \underbrace{\bigcup_{i=1}^{n} \{|_{\preccurlyeq_i^-} c_i^-, c_i^+ |_{\preccurlyeq_i^+}\}}_{P} \right) \\
\vdash_{\exists\forall C} & \underbrace{EQ(\forall y(c^- \preccurlyeq^- y \preccurlyeq^+ c^+ \rightarrow \{b^- \preccurlyeq_b^- \bar{x}+y \preccurlyeq_b^+ b^+\}))}_{I}
\end{array}
$$

with $\theta \in [\![I]\!]$.

For G obtained by translation of $A\,\mathscr{S}_J\,B$,

$$
\begin{aligned}
T_{(P,\mathscr{T})}\uparrow n \models\ & \exists y(\{y<0, y\in J\} \wedge \pi(A, \bar{x}+y, C) \wedge \\
& \qquad \forall y'(y<y'<0 \rightarrow \pi(B, \bar{x}+y', C)))\theta \\
& \text{iff} \\
T_{(P,\mathscr{T})}\uparrow n \models\ & \underbrace{(\pi(A, \bar{x}+y, C) \wedge \forall y'(y<y'<0 \rightarrow \pi(B, \bar{x}+y', C)))\theta\{x \leftarrow b\}}_{\pi(A,\bar{x}+y,C)\theta\{x\leftarrow b\}\ \wedge\ \forall y'(b<y'<0\ \rightarrow\ \pi(B,\bar{x}+y,C)\theta)}
\end{aligned}
$$

for some $b<0$ and $b\in J$. Applying inductive hypothesis, there are $\vdash_{ODMTL}$-derivations

$$\pi(A, \bar{x}+y, C) \vdash_{ODMTL} I_1, \tag{45}$$

$$\forall y'(b<y'<0 \rightarrow \pi(B, \bar{x}+y, C)) \vdash_{ODMTL} I_2, \tag{46}$$

such that $\theta\{x \leftarrow b\} \in [\![I_1]\!]$ and

$$\Pi(P) \models \tilde{\forall} I_2 \rightarrow \forall y'(b<y'<0 \rightarrow \pi(B, \bar{x}+y, C)). \tag{47}$$

Since I_i, $i=1,2$, are TCSs containing only $KI(P\cup G)$ multiples and with $I_i \rhd_{\bar{x}} C$, $I_1 \rhd_{\bar{x}+y} C$, $I_2 \rhd_{\bar{x}+y'} C$, I_1 can be rewritten into

$$C \cup \{y<0,\ y\in J,\ b^- \preccurlyeq_b^- \bar{x}+y \preccurlyeq_b^+ b^+\}$$

and I_2 has been obtained from an I_2' being of the form

$$I_2' = C \cup \{a^- \preccurlyeq_a^- \bar{x}+y' \preccurlyeq_a^+ a^+\} \tag{48}$$

by application of the $(\exists\forall C)$-rule, whereby b^-, b^+, a^-, a^+ are precise bounds computed via $\rightarrow_{pdsc}$. The following holds:

$$|_{\leqslant_b^-} b^-, b^+ |_{\leqslant_b^+} \cap |_{\leqslant_a^-} a^-, a^+ |_{\leqslant_a^+} \neq \emptyset$$

or $b^+ = a^-$ and $\leqslant_b^+ = \leqslant$. Otherwise, one of the following would be true:

1. If $b^+ < a^-$, then there is a $d \in \mathbb{Q}$ with $b^+ < d < a^-$. Hence, due to $\theta(\bar{x})\{y \leftarrow b\} \in [\![I_1]\!]$,

$$\theta\{y \leftarrow b\}(\bar{x} + y) \leqslant_b^+ b^+ < d < a^- \leqslant_a^- \theta\{y' \leftarrow b'\}(\bar{x} + y')$$

   for all $\theta\{y' \leftarrow b'\} \in [\![I_2']\!]$, since (48), in contradiction to (47).
2. If $b^+ = a^-$ and $\leqslant_b^+ = <$, then $\theta(\bar{x}) + \{y \leftarrow b\}(y) < b^+$, since $\theta\{y \leftarrow b\} \in [\![I_1]\!]$, and $b^+ = a^- \leqslant \theta\{y' \leftarrow b'\}(\bar{x} + y')$ for all $\theta\{y' \leftarrow b'\} \in [\![I_2']\!]$. Hence, there is a $d \in \mathbb{Q}$ with

$$\theta(\bar{x}) + \{y \leftarrow b\}(y) < d < \theta\{y' \leftarrow b'\}(\bar{x} + y')$$

   for all $\theta\{y' \leftarrow b'\} \in [\![I_2']\!]$, in contradiction to (47).

Therefore, we have either $b^+ > a^-$ or $b^+ = a^-$ and $\leqslant_b^+ = \leq$. Furthermore,

$$\max(b^-, a^-) < \theta(\bar{x}) \leqslant a^+, \tag{49}$$

since $y' < 0$, $a^- \leqslant \bar{x} + y'$, $y < 0$, $b^- \leqslant \bar{x} + y < \bar{x} + y' \leqslant_a^+ a^+$, and the last inequality holds for all $y' < 0$.

By assumption, (46),

$$\begin{aligned} G = \quad & \exists y(\{y < 0, y \in J\} \wedge \pi(A, \bar{x} + y, C) \wedge \forall y'(y < y' < 0 \rightarrow \pi(B, \bar{x} + y', C)) \\ \vdash_{ODMTL} \ & I_1 \wedge \forall y'(y < y' < 0 \rightarrow \pi(B, \bar{x} + y', C)) \text{ due to (45)} \\ \vdash_{ODMTL} \ & I_1 \wedge \forall y' \left(y < y' < 0 \rightarrow \pi(B, \bar{x} + y', C) \middle| \bigwedge B_i : P \right) \end{aligned}$$

with $|_{\leqslant_a^-} a^-, a^+ |_{\leqslant_a^+} \in \bigcup P$. Because of 1, 2, and (49) the rule $(\forall \mathscr{S} C)$ is applicable and

$$\begin{aligned} & [\![I_1 \cup I_2' \cup \{\max(b^-, a^-) < \bar{x} \leqslant a^+\}]\!]_{Var(\bar{x})} \\ & \qquad = \\ & [\![I_1 \cup \{\max(b^-, a^-) < \bar{x} \leqslant a^+\}]\!]_{Var(\bar{x})} \end{aligned}$$

is satisfiable with $\theta \in [\![I_1 \cup \{\max(b^-, a^-) < \bar{x} \leqslant a^+\}]\!]$. Thus, the derivation above can be continued with

$$\vdash_{\exists\forall \mathscr{S} C} I_1 \cup \{\max(b^-, a^-) < \bar{x} \leqslant a^+\} = I$$

and $\theta \in [\![I]\!]$. $\square$

12. Negation as failure

The calculus *ODMTL* presented in the foregoing section can be extended by negation as failure. This was first observed by Schäfer [97] and utilized as the basis for handling negation within the temporal logic programming system LIMETTE [26].

It relies on the observation that $\rightarrow_{pdsc}$ normalized answer constraints $C \cup \{c^- \preccurlyeq^- \bar{x} \preccurlyeq^+ c^+\}$ for queries $G(\bar{x})$ can be also seen as specifying intervals with bounds c^-, c^+, within which $G(\bar{x})$ holds, since for each c within $c^- \preccurlyeq^- c \preccurlyeq^+ c^+$ there is a solution θ of $c \cup \{c^- \preccurlyeq^- \bar{x} \preccurlyeq^+ c^+\}$ with $\theta(\bar{x}) = c$.

Let us consider goals of the form

$$\exists \bar{x}(C \wedge \ \textit{not}\ G(\bar{x})), \tag{50}$$

which are proven according to the *negation as failure principle* by proving the subgoals

$$\exists \bar{x}(C \wedge G(\bar{x})) \vdash_{ODMTL} C_i \cup \{c_i^- \preccurlyeq_i^- \bar{x} \preccurlyeq_i^+ c_i^+\}.$$

Whenever all such successful derivations for $\exists \bar{x}(C \wedge G(\bar{x}))$ are determined with answer constraints $C_i \cup \{c_i^- \preccurlyeq_i^- \bar{x} \preccurlyeq_i^+ c_i^+\}$ for $i = 1, \ldots, n$, then

$$\Pi(P) \models \tilde{\forall} \left(\bigvee_{i=1}^{n} C_i \cup \{c_i^- \preccurlyeq_i^- \bar{x} \preccurlyeq_i^+ c_i^+\} \right) \rightarrow G(\bar{x})$$

and by *closed world assumption*

$$\Pi(P) \models \tilde{\forall} \neg \left(\bigvee_{i=1}^{n} C_i \cup \{c_i^- \preccurlyeq_i^- \bar{x} \preccurlyeq_i^+ c_i^+\} \right) \rightarrow \neg G(\bar{x}). \tag{51}$$

Eliminating all variables in $C_i \cup \{c_i^- \preccurlyeq_i^- \bar{x} \preccurlyeq_i^+ c_i^+\}$ except those in $\bar{x}$ we obtain

$$\bigvee_{i=1}^{n} (C \cup \{c_i^- \preccurlyeq_i^- \bar{x} \preccurlyeq_i^+ c_i^+\}) \leftrightarrow C \wedge \bigvee_{i=1}^{n} \{c_i^- \preccurlyeq_i^- \bar{x} \preccurlyeq_i^+ c_i^+\}. \tag{52}$$

The implication (51) simplifies therefore to

$$\tilde{\forall} \left(\neg C \vee \neg \bigvee_{i=1}^{n} \{c_i^- \preccurlyeq_i^- \bar{x} \preccurlyeq_i^+ c_i^+\} \right) \rightarrow \neg G(\bar{x}), \tag{53}$$

which can be further simplified into

$$\tilde{\forall} \neg \bigvee_{i=1}^{n} \{c_i^- \preccurlyeq_i^- \bar{x} \preccurlyeq_i^+ c_i^+\} \rightarrow \neg G(\bar{x}), \tag{54}$$

since by assumption, (50), C holds. The bounds within $\{c_i^- \preccurlyeq_i^- \bar{x} \preccurlyeq_i^+ c_i^+\}$ are precise such that

$$\bigvee_{i=1}^{n} \{c_i^- \preccurlyeq_i^- \bar{x} \preccurlyeq_i^+ c_i^+\}$$

can be also represented as union of intervals

$$C = \bigcup_{i=1}^{n} |_{\preccurlyeq_i^-} c_i^-, c_i^+ |_{\preccurlyeq_i^+},$$

and $\neg\bigvee_{i=1}^{n}\{c_i^- \leqslant_i^- \bar{x} \leqslant_i^+ c_i^+\}$ as complement $\bar{C}$ of C with respect to $\mathbb{Q}$ (respectively, $\mathbb{Z}$), which again can be represented by a union of intervals

$$\bigcup_{i=1}^{n} |_{b_i^-} b_i^-, b_i^+ |_{b_i^+}.$$

(54) simplifies thereby into

$$\tilde{\forall} \bigvee_{i=1}^{n} \{b_i^- \leqslant_{b_i}^- \bar{x} \leqslant_{b_i}^+ b_i^+\} \rightarrow \neg G(\bar{x}).$$

Inference rules formalizing the methods sketched above are given in Fig. 21. The method itself is a specialization of *constructive negation* for constraint logic programs [98] but it avoids explicit handling of $\forall$-quantified variables.

13. Extensions

The class of bounded universal Horn formulae is the greatest Horn fragment of the metric temporal logic considered having all the properties assumed to be essential for a logic programming language.

The extension by $\Diamond$ operators in heads leads to the loss of least models, as in the disjunctive logic programming case, since $\Diamond_c p$ represents a disjunction $\bigvee_{i=0}^{c} \circ^i p$ (respectively, $\Diamond_I p$ represents $\bigvee_{c \in I} \Diamond_{[c,c]} p$). The relaxation of constant bounds for $\Box_c$ (respectively, $\Box_I$) operators in bodies results in case of unbounded $\Box$ into an incomplete fragment, since proving of $\Box\Diamond A$ formulae from simple MTL-programs, even from *Templog* programs (cf. Section 15), is a Σ_1^1-complete problem [81, 40]. [13]

The inclusion of pairs of operators $\circ^x, \bullet^x$ (holds exactly at a distance of x from now), $\Diamond_+^x, \Diamond_-^x$ (holds sometime within a distance of x from now), or $\Box_+^x, \Box_-^x$ (holds always till a distance of x form now) in MTL-Horn formulae and MTL-goals leads to a logic that has the full expressive power of linear arithmetical constraints over the time structure under investigation. [14] Such operators with variable bounds, however, are used in [69] and are essential for the real-time logics proposed in [8, 6, 53].

Theorem 13.1. *Let X be a bounded universal Horn program and G a bounded universal goal G containing pairs of operators $\circ^x, \bullet^x$ (respectively, $\Diamond_+^x, \Diamond_-^x$ or $\Box_+^x, \Box_-^x$). Then $X \models G$ is as hard as $X' \models G'$ for a constraint logic program X' and goal G' with linear inequalities over the underlying time domain as the constrain domain.*

[13] This result can be shown using a standard encoding of a nondeterministic Turing machine by a set of Horn formulae modeling computation steps by progress in time, and expressing a Σ_1^1-complete problem [54] – whether a given nondeterministic Turing machine has a computation over an empty tape repeating its starting state infinitely often – as a $\Box\Diamond A$ sentence for A modeling the fact that the computation of the Turing machine is in its starting state. Using a definition of the $\Diamond$ operator – $\Diamond A \leftrightarrow (A \vee \circ \Diamond A)$ – by a set of temporal Horn formulae, this property can be also expressed by an $\Box A$ formula.

[14] In [70] the operator $\circ^t$ is denoted by $\Diamond_{=t}$.

Proof. We can encode a set of linear inequalities

$$\begin{array}{l} a_{11}x_1+\cdots+a_{1n}x_n\leqslant b_1 \\ \qquad\vdots \\ a_{m1}x_1+\cdots+a_{mn}x_n\leqslant b_m \end{array} \tag{55}$$

by the set P of (simple) MTL-Horn formulae $P=$

$$\begin{array}{lll} p & \leftarrow & \underbrace{\circ^{x_1}\cdots\circ^{x_1}}_{a_{11}}\cdots\underbrace{\circ^{x_n}\cdots\circ^{x_n}}_{a_{1n}}q_1\wedge \\ & & \qquad\vdots \\ & & \underbrace{\circ^{x_1}\cdots\circ^{x_1}}_{a_{m1}}\cdots\underbrace{\circ^{x_n}\cdots\circ^{x_n}}_{a_{mn}}q_m \\ \Box_{b_i}\Box_{-}q_i & & (i=1,\ldots,m). \end{array}$$

Negative coefficients a_{ij} in the formulae above are encoded by $\underbrace{\bullet^{x_j}\cdots\bullet^{x_j}}_{a_{ij}}$. p follows from P iff (55) is solvable. The inequalities (55) can be also expressed using $\Diamond_+^x$ and $\Diamond_-^x$ (respectively, $\Box_+^x$ and $\Box_-^x$) as can be seen at the following programs:

$$\begin{array}{lll} p & \leftarrow & \underbrace{\Diamond_-^{x_1}\ldots\Diamond_-^{x_1}}_{a_{11}}\ldots\underbrace{\Diamond_-^{x_n}\ldots\Diamond_-^{x_n}}_{a_{1n}}q_1\wedge \\ & & \qquad\vdots \\ & & \underbrace{\Diamond_-^{x_1}\ldots\Diamond_-^{x_1}}_{a_{m1}}\ldots\underbrace{\Diamond_-^{x_n}\ldots\Diamond_-^{x_n}}_{a_{mn}}q_n \\ \circ^{-b_i}\Box_+q_i & & (i=1,\ldots,m), \end{array}$$

respectively,

$$\begin{array}{lll} p & \leftarrow & \underbrace{\Box_+^{x_1}\ldots\Box_+^{x_1}}_{a_{11}}\ldots\underbrace{\Box_+^{x_n}\ldots\Box_+^{x_n}}_{a_{1n}}q_1\wedge \\ & & \qquad\vdots \\ & & \underbrace{\Box_+^{x_1}\ldots\Box_+^{x_1}}_{a_{m1}}\ldots\underbrace{\Box_+^{x_n}\ldots\Box_+^{x_n}}_{a_{mn}}q_n \\ \Box_{[-\infty,b_i]}\,q_i & & (i=1,\ldots,m). \qquad \Box \end{array}$$

In the discrete case, this gain of expressiveness has two negative consequences. Firstly, satisfiability checking during MTL-derivations has to cope with general linear inequalities over the integers, which satisfiability is well-known as being a NP-complete problem. The fragment of simple MTL-programs is thus equivalent to constraint logic programs with linear inequalities over the integers. Secondly, the proof method for bounded universal Horn formulae presented in this paper becomes incorrect for temporal operators with variable bounds. To see this, consider the program

$$\circ^x \circ^x \circ^x p \leftarrow \;\equiv_\Pi\; p(x+x+x) \leftarrow$$

and the goal

$$\leftarrow \Box_3 p \equiv_\Pi \;\leftarrow \forall y(0 \leqslant y \leqslant 3 \rightarrow p(y)).$$

Its derivation via $\vdash_{EMTL}$ yields $\forall y(0 \leqslant y \leqslant 3 \rightarrow 0 \leqslant 3x - y \leqslant 0)$ to be eliminated with the LR-lemma, which, however, as has been remarked below Lemma 7.8, does not hold in this case.

In the dense case, proving queries of the form $\Box A$, for A being an atomic formula, from (dense) bounded universal Horn formulae forms a Σ^1_1-complete problem as well. Contrary to the discrete case, however, also inclusion of temporal operators with variable bounds leads already to Σ^1_1-hardness of the corresponding consequence problem.

13.1. A Σ^1_1-complete problem

The problem whether a given nondeterministic Turing machine has an infinite recurring computation is well known to be Σ^1_1-complete [54].

Lemma 13.2. *The problem of deciding whether a given nondeterministic Turing machine has, over the empty tape, a computation in which the starting state is visited infinitely often, is Σ^1_1-complete.*

Now we proceed by encoding a nondeterministic Turing machine by a set of bounded universal Horn formulae with variable bounds and use the encoding to prove Σ^1_1-hardness of the corresponding consequence relation.

Theorem 13.3. *Given a set of bounded universal Horn formulae with variable bounds X and a bounded universal goal G, the problem $X \models G$ is Σ^1_1-hard.*

Proof. Given a nondeterministic Turing machine M with alphabet V, states Q, and transition function $\delta : V \times Q \times V \rightarrow 2^{(V \cup S)^3}$ such that a configuration $c = x\sigma q\tau z$, for $x \in V^*$, $\sigma, \tau \in V$ and $q \in Q$, can result in a configuration $xy^R z$ for each $y^R \in \delta(\sigma, q, \tau)$.

We define a set of bounded universal Horn formulae P with variable bounds and a goal G over a signature $\Sigma = (S, F, P)$ with $S = \{tape\}$, unary function symbols to be used for the encoding of the alphabet $F = \{a : tape \rightarrow tape \mid a \in V\} \cup \{\flat\}$, and predicate symbols $P = \{q : tape\ tape \mid q \in Q\}$ encoding the states of M. A configuration $c = \flat\sigma_1 \ldots \sigma_n q \tau_1 \ldots \tau_m \flat$ is represented by a predicate $q(\sigma_n(\ldots\sigma_1(\flat)\ldots), \tau_1(\ldots\tau_m(\flat)\ldots))$, the initial configuration by $q_0(\flat, \flat)$.

The transition functions δ is coded by a set X of bounded universal Horn formulae. It contains a Horn formula for each entry of the transition table defined as follows:

$$\begin{aligned}
&\Box(q'(x,\sigma(\tau'(y))) \leftarrow q(\sigma(x),\tau(y))) \\
&\qquad\qquad\qquad\text{if } q'\sigma\tau' \in \delta(\sigma,q,\tau),\ q \neq q_0, \\
&\Box(q'(\sigma(x),\tau'(y)) \leftarrow q(\sigma(x),\tau(y))) \\
&\qquad\qquad\qquad\text{if } \sigma q'\tau' \in \delta(\sigma,q,\tau),\ q \neq q_0, \\
&\Box(q'(\tau'(\sigma(x)),y) \leftarrow q(\sigma(x),\tau(y))) \\
&\qquad\qquad\qquad\text{if } \sigma\tau' q' \in \delta(\sigma,q,\tau),\ q \neq q_0, \\
&\Box(\Box_{[x,1]}q'(y,\sigma(\tau'(z))) \leftarrow \circ^x\circ^x q_0(\sigma(y),\tau(z))) \\
&\qquad\qquad\qquad\text{if } q'\sigma\tau' \in \delta(\sigma,q_0,\tau) \\
&\Box(\Box_{[x,1]}q'(\sigma(y),\tau'(z)) \leftarrow \circ^x\circ^x q_0(\sigma(y),\tau(z))) \\
&\qquad\qquad\qquad\text{if } \sigma q'\tau' \in \delta(\sigma,q_0,\tau) \\
&\Box(\Box_{[x,1]}q'(\tau'(\sigma(y)),z) \leftarrow \circ^x\circ^x q_0(\sigma(y),\tau(z))) \\
&\qquad\qquad\qquad\text{if } \sigma\tau' q' \in \delta(\sigma,q_0,\tau).
\end{aligned}$$

The problem whether M repeats its starting state – q_0 – infinitely often can then be expressed as

$$\Box(s \leftarrow q_0(x,y))$$

$$\circ q_0(\flat,\flat)$$

and the goal $\Box_{(0,1]}s$.

Clearly, M starting on an empty tape, has an infinite computation which repeats its starting state infinitely often iff $X \models \Box_{(0,1]}s$.

The encoding given above can be also based (with slight modifications) on $\Box^x_+$, $\Box^x_-$ (respectively, on $\Diamond^x_+$, $\Diamond^x_-$) operators instead of $\circ^x$, $\bullet^x$. $\Box$

14. Anchored flows of time

So far we have considered models with time structures isomorphic to the integers and to the rational numbers. The results of this paper, however, hold also for anchored versions of these time structures, i.e. for the natural number and positive rational flow of time. They can be easily obtained by modifying the translation of atomic formulae in Section 4 to

$$\pi(p(r_1,\dots,r_n),t) = p(t,r_1,\dots,r_n) \wedge \{t \geqslant 0\}$$

occurring in goals and bodies, and to

$$\pi(p(r_1,\dots,r_n)) = \{t \geqslant 0\} \rightarrow p(t,r_1,\dots,r_n)$$

for atomic formulae in heads of bounded universal formulae. The semantical characterizations remain valid along with $\vdash_{EMTL}$ (respectively, $\vdash_{DEMTL}$ and $\vdash_{ODMTL}$) and their complexity bounds, since t is a path starting from the root of the tree associated to the formulae being translated.

15. Related work

15.1. Temporal logic programming

Templog, the most extensively investigated temporal logic programming language, proposed in [2] and investigated in [15, 21, 17, 16] is a fragment of simple MTL-programs that contains only $\Box_+, \circ, \Diamond_+$ operators. The execution mechanism of Templog, the TSLD-resolution, is based on the equivalences $\circ\Diamond_+A \leftrightarrow \Diamond_+\circ A$, $\circ(A \wedge B) \leftrightarrow \circ A \wedge \circ B$, $\Diamond_+(\Diamond_+A_1 \wedge \cdots \wedge \Diamond_+A_n) \leftrightarrow \Diamond_+A_1 \wedge \cdots \wedge \Diamond_+A_n$ allowing to simplify each Templog goal into a goal of the form $\Diamond_+(B' \wedge \circ^i A \wedge B'')$ or $\leftarrow B' \wedge \circ^i A \wedge B''$ for some atomic formula A. Such normal forms of goals, however, do not always exists. In fact, (simple) MTL-goals cannot be rewritten into goals with a bounded nesting of sometime operators as can be seen on the following formulae A_i^+, A_i^+ defined by $A_0^+ = \Diamond_+A^+$, $A_0^- = \Diamond_-A^-$, $A_{n+1}^+ = \Diamond_+(A_n^+ \wedge A_n^-)$, and $A_{n+1}^- = \Diamond_-(A_n^+ \wedge A_n^-)$, where A^+, A^- are distinct atomic formulae (except by renamings techniques [41]).

The languages investigated in [46, 48] are based on a different subset of temporal operators. D. Gabbay studied a variety of implication languages in the uniform framework of *labeled deduction systems*. The simplest one is very close to Horn logic while the most expressive one covers full temporal logic. Contrary to the fragment considered in this paper, his language is based on sometime operators – $\Diamond_+$ and $\Diamond_-$ – which can occur in heads and bodies of implications. MTL-clauses of the form $\Box(A \to \Box B)$ are not allowed there as clauses but clauses of the form $\Box(B \to \Diamond_+A)$. He studied therefore primarily the problem of handling Skolem functions introduced by $\Diamond_+$- and $\Diamond_-$-operators in the heads of implications and proof methods for dealing with them. The issue of efficiency was not studied in the paper.

BNR-Prolog and *Starlog* [32] use interval arithmetics over the reals with $+, *, =, \leqslant$ to model intervals and to describe temporal properties in logic programs. In [95] a subset of IQ-logic developed by Richards is used as a basis for a temporal logic programming language IQ-Prolog. The language uses a number of temporal operators indexed by terms specifying their temporal range. The operational semantics of the language is given by a translation into a constraint logic programming language with linear constraints over the time domain considered. Since the temporal operators include variables the complexity of satisfiability checking during CLP-derivations should be as high as the complexity of solving linear constraints (cf. Section 13). In a recent work [44] proposed to utilize the framework of *annotated constraint logic programming* for temporal reasoning in logic programming and discussed its realization using special constraint theories. This framework is similar to that of constraint logic programming and covers only temporal properties expressible without nesting of temporal operators, i.e., as in $\Box_{c'} \Diamond_c A$ or $\Diamond_c(A \wedge \Diamond_{c'} B)$, but allows to express existential temporal properties in heads of temporal Horn formulae. No complexity results for derivations within this calculus has been presented.

Furthermore, Chomicki and Imieliński considered a temporal extension of Datalog (logic programming without function symbols) obtained by tagging each predicate with an additional argument modeling time. Due to the restrictions to one monadic function symbol modeling time and to constants and variables modeling data they obtained a decidable Horn logic. The expressive power of the language coincides with Templog without function symbols [17].

Temporal extensions of first-order Horn logic have been investigated in [56, 57, 92]. Hrycej [56, 57] bases his work on a fragment of Allen's temporal logic [4, 5] with a Horn logical axiomatization which is used as basis for the operational semantics of his *Temporal Prolog*. In his approach the consistency of expressions of Allen's time algebra is checked for efficiency reasons by an incomplete version of Allen's algorithm for satisfiability checking [4]. The language itself can be seen as a CLP language over Allen's time algebra [4]. A. Porto and Cristina Ribeiro [92] proposed an interval temporal logic *MI* for knowledge based systems described by temporal Horn clauses. They consider a language with functions in data bases (i.e., programs) with partially specified temporal relations and study the problem of consistent completion of these temporal relations in order to prove a given goal. They presented a proof system for bottom-up computation of the language but without soundness and completeness results.

In [34] the results and techniques developed in the context of automated theorem proving in modal logics by *functional* translations into first-order logic have been applied to modal Horn logics. They defined a language called PATHLOG on the level of translated modal Horn formulae, in which modal Horn formulae are mapped. The properties of function symbols introduced by these translations reflect on the first-order level the properties of the accessibility relations of the frames of the corresponding modal logics such that proving in (some) modal logics can be reduced to theorem proving modulo equational theories, for which unification algorithms are known [12]. [34] characterized the terms introduced by the translations and showed that unification of these terms in case of the modal logic KD4 leads to *finitary* unification problems although the underlying equational theory of associativity is infinitary in terms of the unification hierarchy [12]. [15] They also mapped Templog formulae into PATHLOG formulae modulo associativity of the function symbols introduced by the translation and reduced thereby proving of Templog goals from Temporal programs to SLD-derivation modulo associativity.

Model-theoretic and fixed point semantics for modal and intensional Horn logics have been studied in [13, 89]. While [13] and also [36, 14] are more interested in classical modal logics like T and S4, the framework of [89] can be also applied to temporal

[15] This has been also shown in [86].

languages like Templog. The results of [89] are, however, not sufficient to obtain the semantical characterizations for the class of bounded universal Horn formulae.

A completely different approach to *temporal logic programming* is taken in [100, 84, 51, 47, 18, 45, 82]. Contrary to the logic programming paradigm, which sees program execution as deduction in computationally tractable fragments of suitable logics, program execution is considered there as construction of Kripke models for the program formulae. The main motivation of this line of research is to provide a logical basis for the specification, verification and execution of imperative programs [100, 84, 51, 82], for the combination of logic and imperative programming [45, 47], and for programming of *reactive* systems [47, 18].

In [47] D. Gabbay considered a fragment of temporal logic for programming with $\mathscr{S}$ operators in goals that forms the theoretical basis for the MetateM system. Besides the programmatic differences – he uses temporal logic as a basis for the integration of imperative and logic programming – he also addressed the problem of proving goals with $\mathscr{S}$ operators. His method unfolds queries with $\mathscr{S}$ operators using the equivalence $A \mathscr{S} B \leftrightarrow \bullet(B \vee (A \wedge (A \mathscr{S} B)))$ of linear discrete temporal logic and tries to prove the B part of the disjunction by standard methods. In case of failure he tries to prove the recursive part of the disjunction. This method is a variation of the unfolding approach sketched in Section 6 and is not able to exploit uniform proofs. A recent survey on temporal logic programming is given in [88].

15.2. Theorem proving in modal and temporal logics

This work has been inspired by the success of the theorem proving methods for modal logics relying on so called functional translations into classical logic [106, 86, 37, 11, 85, 43], which can be traced back to [106]. He presented a proof method for several (classical) modal logics based on a translation into classical logic and specialized algorithms for checking modal dependencies for the connection method of W. Bibel (respectively, the matrix method of P.B. Andrew). This idea has been then applied to resolution based methods and further developed [86, 37, 11, 85, 43]. The novelty of the approach is to translate modal formulae into formulae of classical logic such that reachability described by modal operators is reflected on the level of terms produced by the translation. More precisely, reachability with respect to the relation underlying the considered class of Kripke-structures is mirrored in the properties of the function symbols generated by the translation. For some of the (classical) modal logics, these properties can be described by equational theories, which have been investigated in the context of the *unification theory* such that already known unification algorithms could be used to check inhabitance in the same "possible world". For a short, historical sketch of the development see [87], a more methodical description can be found in [43].

Unfortunately, this technique can be applied only to rather simple modal logics, for which classes of Kripke frames can be axiomatized by a set of first-order axioms. In general, however, modal axiom schema correspond to higher-order axioms of classical logic [104], especially in the case of temporal logics over inductive time structures as investigated in this paper, which have no first-order equivalents. This is also reflected by the fact that full first-order (temporal) logics are incomplete in general [99, 1].

The techniques introduced and developed in this paper, can be seen as a further development of the methods presented in the functional translation context. A characterization of terms introduced by the translation as having the so called *prefix stability property* was already given in [86] (respectively, *unique prefix property* in [34]). This property ensures that unification problems arising during derivations of translated modal formulae are finitary only, although unification under *associativity*, which is the underlying theory in case of transitive Kripke frames, is infinitary in general [12].

The contribution of this work in view of this development lies in the characterization of complete fragments of first-order temporal logics, development of elimination algorithms for the quantifiers introduced by the translation – a fragment of the theory of real arithmetic (respectively, of the Presburger arithmetics), and to generalize SLD-resolution, including additional operations manipulating terms – sets of linear inequalities – coding temporal dependencies, in order to obtain a complete proof method.

An attempt to utilize constraint based proving methods for temporal reasoning in a first-order framework has been also undertaken in [91], who presented an extension of constraint resolution [30] allowing to reason about intervals. He bases his work on a (first-order) temporal logic with explicit time points, intervals, and explicit functions and relations. Due to the framework, which relies upon open predicate logic, the expressiveness of the logic is not sufficient to express properties involving alternation of quantification, e.g. $\Box_{20}\Diamond A$, and avoids thereby the quantifier elimination. Furthermore, the unrestricted usage of linear inequalities yields already in case of discrete time structures to NP-complete constraint satisfaction problems.

15.3. Temporal databases

As already mentioned in the previous section J. Chomicki and T. Imieliński presented an extension of Datalog, $Datalog_{+1}$, by an unary function symbol, which coincides with the function-free fragment of Templog, for the representation of (deductive) temporal databases. This fragment has been then further investigated with respect to expressiveness and complexity question (cf. [17] for an overview).

The work on $Datalog_{+1}$ can be seen as a predecessor of the work on constraint databases [65] which generalize the notation of tuple data types to conjunctions of constraints of an appropriate language. In their fundamental paper [65], Kannalakis et al. considered besides Boolean Algebra, the theory of dense linear order and that of equality also the theory of real-closed fields as the constraint theory of the underlying query language. Our work on the level of translated temporal logic programs can be seen as a characterization of a fragment of the theory of real arithmetic (respectively, Presburger arithmetic) which admits more efficient quantifier elimination techniques than those developed for the whole theory. The problem itself is known to have double exponential time complexity for the theory of Presburger arithmetic [42] (respectively, nondeterministic exponential time complexity for the theory of real arithmetic [42]).

In contrast to that, the quantifier elimination problems, which arise during evaluation of bounded universal Horn programs can be solved in linear time (in the number of variables). The expressiveness of the query language as defined by the bounded universal Horn formulae fragment is however that of full first-order temporal logic with negation interpreted as negation as failure. [16]

Query languages for temporal databases have been also presented in [102, 63]. Both approaches are based on an extended relational algebra. The first uses an extension by *linear recursion operator*, the second relies on constraints on *linear repeating points* allowing to express periodic temporal informations. Linear repeating points are restricted expressions of Presburger arithmetics which can, in principle, exploit the structural temporal information being implicit in the considered *life span representations*. Elimination of universal quantifiers, however, leads also to exponential time complexity of the method.

Acknowledgements

I would like to thank A. Bockmayr, P. Deussen, M. Gollner, G. Goos, T. Käufl, K. Schäfer, C. Schulte, G. Thimm, and A. Werner for interesting and instructive discussions on temporal logic and various aspects of logic programming and constraint solving.

Appendix A. Notation

We assume the reader is familiar with the theory of logic programming and constraint logic programming (cf. [59]) and introduce only briefly our notational conventions. A *signature* Σ is a triple (S,F,P) consisting of a set S of sorts, a set $F = \bigcup_{w \in S^*, s \in S} F_{w,s}$ of function symbols, and of a set $P = \bigcup_{w \in S^*} P_w$ of predicate symbols. We write $f : s_1 \ldots s_n \to s$ if $f \in F_{s_1 \ldots s_n, s}$ and $p : s_1 \ldots s_n$ if $p \in P_{s_1 \ldots s_n}$. $Var(t)$ (respectively, $Var(A)$) denotes the set of variables occurring in a term t (respectively, in a formula A), $\mathcal{T}_\Sigma(\mathcal{V}) = \bigcup_{s \in S} \mathcal{T}_\Sigma(\mathcal{V})_s$ denotes terms over a signature Σ and a set of variables $\mathcal{V}$, and $\mathcal{T}_\Sigma = \bigcup_{s \in S} \mathcal{T}_{\Sigma s}$ denotes ground terms over Σ. An *atom* is of the form $p(t_1, \ldots, t_n)$, if $p : s_1 \ldots s_n$ and $t_i \in \mathcal{T}_\Sigma(\mathcal{V})_{s_i}$.

A (Σ-)*substitution* is a mapping $\sigma : \mathcal{V} \to \mathcal{T}_\Sigma(\mathcal{V})$ that is the identity except for finitely many variables and that satisfies $\sigma(\mathcal{V}_s) \subseteq \mathcal{T}_\Sigma(\mathcal{V})_s$ for all $s \in S$. We write substitutions as

[16] The query expressiveness of the language with respect to a model defined by a temporal logic program is that of full first-order logic since $\mathcal{S}$ and $\mathcal{U}$ are expressively (functionally) complete (over Dedekind complete flow of time, i.e. including integers and reals but not rationals) [64] and in case of formulae evaluation in a given model there is no distinction between negation and negation as failure.

finite set of variable replacements $\{x_1 \leftarrow t_1, \ldots, x_n \leftarrow t_n\}$. The *domain of a substitution* σ is defined by $dom(\sigma) = \{x \mid \sigma(x) \neq x\}$. An *unifier* of two terms t, t' is a substitution σ such that $\sigma(t) = \sigma(t')$; a unifier θ is called *most general* (*mgu*) if for any unifier σ of t, t' there exists a substitution λ such that $\sigma = \lambda \circ \theta$. We often use postfix notation for application and composition of substitutions, i.e. we write $t\sigma$ and $\sigma\lambda$ for $\sigma(t)$ and $\lambda \circ \sigma$, respectively. A (first-order) Σ-structure $\mathcal{M}$ is a triple $(\mathcal{D}, (f^{\mathcal{M}})_{f \in F}, (p^{\mathcal{M}})_{p \in P})$ with $\mathcal{D} = \bigcup_{s \in S} \mathcal{D}_s$ and $\mathcal{D}_s \neq \emptyset$ for all $s \in S$. An assignment into a (first-order) Σ-structure $\mathcal{M} = (\mathcal{D}, (f^{\mathcal{M}})_{f \in F}, (p^{\mathcal{M}})_{p \in P})$ is a mapping $\alpha : \mathcal{V} \to \mathcal{D}$ with $\alpha(\mathcal{V}) \subseteq \mathcal{D}_s$ for all $s \in S$. Its homomorphic extension to the set of terms $\mathcal{T}_\Sigma(\mathcal{V})$ is denoted by $\overline{\alpha}$.

Within our work we use several simplification relations. As usual, for a relation $\rightarrow$, $\xrightarrow{+}$ denotes its transitive and $\xrightarrow{*}$ its reflexive and transitive closure. $\mathbb{Z}$ denotes the set of integers, $\mathbb{N}$ the natural, $\mathbb{Q}$ the rational, and $\mathbb{R}$ the real numbers. We use extensions of these sets with $-\infty$ and ∞, for which the ordering $<$ (on those sets) is extended by $-\infty < c < \infty$ for $c \in \mathbb{Z}$ (respectively, $c \in \mathbb{N}$, $c \in \mathbb{Q}$, or $c \in \mathbb{R}$). Barred variables $\bar{x}$ denote sum terms of the form $x_1 + \ldots + x_n$, bold variables $\mathbf{x}$ tuples of variables $x_1, \ldots, x_n$, bold terms $\mathbf{r}$ tuples of terms $r_1, \ldots, r_n$.

Appendix B. Completeness of the $\rightarrow_{dqsc}$-simplification

In order to show completeness of the $\rightarrow_{dqsc}$-simplification, we generalize the notion of tree constraint systems to that of *quasi tree constraint systems*, which may also contain inequalities for suffixes of paths of the underlying tree.

A set of inequalities C is called a *Quasi Tree Constraint System* or short a *QTCS* with respect to a tree $(T, \leqslant)$ if C is of the form

$$C \subseteq \{c^- \preccurlyeq \bar{y} \preccurlyeq^+ \mid \bar{y} \text{ is a suffix of a path starting from the root } \bar{x} = x_1 + \cdots + x_n \text{ in } (T, \leqslant)\text{, i.e. } \bar{y} = x_i + \cdots + x_n \text{ for some } i\}$$

for $c^-, c^+ \in \mathbb{Q}$. The $\rightarrow_{dsc}$-simplification rules need to be generalized only slightly to be complete for quasi tree constraint systems as well (Fig. 13).

Theorem B.1 (Completeness). *Let C denote a QTCS. Then the following holds:*
(Invariance) *If $C \rightarrow_{dqsc} C'$, then $[\![C]\!] = [\![C']\!]$.*
(Completeness) *If C is unsatisfiable, then $C \xrightarrow{*}_{dqsc} C'$ for some C' containing an inequality $c_1 \preccurlyeq_1 \bar{x} \preccurlyeq_2 c_2$ such that either*

1. *$c_1 > c_2$ or*
2. *$\preccurlyeq_i = <$ for some i and $c_1 = c_2$.*

Proof. We show that inference rules (QILB) and (QIUB) simulate variable elimination according to Fourier's algorithm eliminating variables being leaves of the underlying tree, which shows both the invariance and completeness part of the theorem.

Let C be a QTCS and y a variable being a leave of the tree underlying C. Assume C has w.l.o.g the form

$$\begin{array}{lll}
\bar{x}_1 + y \preccurlyeq_1^+ c_1^+ & c_1^- \preccurlyeq_1^- \bar{x}_1 + y & d_1^- \preccurlyeq_1^{d^-} F_1(\bar{x}) \preccurlyeq_1^{d^+} d_1^+ \\
\vdots & \vdots & \vdots \\
\underbrace{\bar{x}_m + y \preccurlyeq_m^+ c_m^+ \quad c_m^- \preccurlyeq_m^+ \bar{x}_m + y} & & d_r^- \preccurlyeq_r^{d^-} F_r(\bar{x}) \preccurlyeq_r^{d^+} d_r^+. \\
y \preccurlyeq_1^+ c_1^+ - \bar{x}_1 & c_1^- - \bar{x}_1 \preccurlyeq_1^- y & \\
\vdots & \vdots & \\
y \preccurlyeq_m^+ c_m^+ - \bar{x}_m & c_m^- - \bar{x}_m \preccurlyeq_m^- y &
\end{array}$$

Elimination of y leads then to

$$c_i^- - c_j^+ \preccurlyeq_i^- \downarrow \preccurlyeq_j^+ \bar{x}_i - \bar{x}_j \ (i=1,\ldots,m; j=1,\ldots,m) \tag{56}$$

and

$$\begin{gathered}
d_1^- \preccurlyeq_1^{d^-} F_1(\bar{x}) \preccurlyeq_1^{d^+} d_1^+ \\
\vdots \\
d_r^- \preccurlyeq_r^{d^-} F_r(\bar{x}) \preccurlyeq_r^{d^+} d_r^+.
\end{gathered}$$

Since C is a QTCS, the $\bar{x}_k$ are suffixes of paths starting from the root of C to the leave y, and $\bar{x}_i - \bar{x}_j$ are suffixes of paths leading to ancestors of y in C.

1. $\bar{x}_j$ is a suffix of $\bar{x}_i$, that is

$$\bar{x}_i = \bar{y} + \bar{x}_j \quad \text{and} \quad \bar{x}_i + y = \bar{y} + \bar{x}_j + y$$

for some $\bar{y}$. Application of the (GILB)-rule produces therefore the inequality

$$c_i^- - c_j^+ \preccurlyeq_i^- \downarrow \preccurlyeq_j^+ \bar{y} = \bar{x}_i - \bar{x}_j.$$

2. $\bar{x}_i$ is a suffix of $\bar{x}_j$, that is

$$\bar{x}_j = \bar{y} + \bar{x}_i \quad \text{and} \quad \bar{x}_j + y = \bar{y} + \bar{x}_i + y$$

for some $\bar{y}$. Application of the (GIUB)-rule leads then to the inequality

$$c_i^- - c_j^+ \preccurlyeq_i^- \downarrow \preccurlyeq_j^+ \bar{x}_j - \bar{x}_i.$$

Hence, each inequality in (B.1) produced by the elimination of the variable y can be also obtained by applications of the rules (GILB) and (GIUB) to appropriate inequalities in C. □

References

[1] M. Abadi, The power of temporal proofs, Theoret. Comput. Sci. 65 (1989) 35–83.

[2] M. Abadi, Z. Manna, Temporal logic programming, J. Symbolic Comput. 8 (1989) 277–295.

[3] M. Abadi, Z. Manna, Nonclausal deduction in first-order temporal logic, J. Assoc. Comput. Machinery 37 (2) (1990) 279–317.

[4] J.F. Allen, Maintaining knowledge about intervals, Comm. ACM 26 (11) (1983) 832–843.

[5] J.F. Allen, Towards a general theory of action and time, Artif. Intell. 23 (2) (1984) 123–154.

[6] R. Alur, T.A. Henzinger, Real-time logics: complexity and expressiveness, in: Proc. 5th IEEE Symp. on Logic in Computer Science, Philadelphia, IEEE Computer Soc. Press, Silver Spring, MD, 1990.
[7] R. Alur, T.A. Henzinger, Real-time logics: complexity and expressiveness, Inform. and Comput. 104 (1993) 35–77.
[8] R. Alur, T.A. Henzinger, A really temporal logic, J. Assoc. Comput. Machinery 41 (1) (1994) 181–204.
[9] K.R. Apt, M.H. Emden, Contributions to the theory of logic programming, J. Assoc. Comput. Machinery 29 (1982) 841–862.
[10] E.A. Aschrofft, W.W. Wadge, Intermittent assertion proofs in lucid, in: IFIP, Toronto, 1977.
[11] Y. Auffray, P. Enjalbert, Modal theorem proving: an equational viewpoint, in: Proc. 11th Internat. Joint Conf. on Artificial Intelligence, vol. 1, Detroit, 1989.
[12] F. Baader, J. Siekmann, Unification theory, in: D.M. Gabbay, C.J. Hogger, J.A. Robinson (Eds.), Handbook of Logic in Artificial Intelligence and Logic Programming. Oxford University Press, Oxford, 1993.
[13] Ph. Balbiani, L. Farinas Del Cerro, A. Herzig, Declarative semantics for modal logic programs, in: Proc. Internat. Conf. on Fifth Generation Computer Systems, 1988.
[14] P. Balbiani, A. Herzig, M. Marques, TIM: The Toulouse inference machine, in: H. Boley, M.M. Richter (Eds.), Processing Declarative Knowledge – International Workshop PDK'91, Lecture Notes in Artificial Intelligence, vol. 567, Springer, Berlin, 1991.
[15] M. Baudinet, Temporal logic programming is complete and expressive, in: Proc. 6th Annual ACM Symp. on Principles of Programming Languages, Austin, Texas, 1989.
[16] M. Baudinet, On the expressiveness of temporal logic programming, Inform. and Comput. 117 (1995) 157–180.
[17] M. Baudinet, J. Chomicki, P. Wolper, Temporal deductive databases, in: A. Tansel, J. Clifford, S. Gadia, S. Jajodia, A. Segev, R. Snodgrass (Eds.), Temporal Databases, Benjamin/Cummings, Redwood City, Menlo Park, CA, 1993, Ch. 13, pp. 294–320.
[18] H. Barringer, M. Fisher, D. Gabbay, G. Gought, R. Owens, A framework for programming in temporal logic, in: Stepwise Refinement of Distributed Systems, Models, Formalisms, Correctness, REX Workshop, Mook, Netherlands, 1989, Lecture Notes in Computer Science, vol. 430, Springer, Berlin.
[19] W. Bibel, Prädikatives Programmieren, in: H. Brakhage, (Ed.), Automata Theory and Formal Languages, 2nd GI Conf., Lecture Notes in Computer Science, vol. 33, Springer, Berlin, 1975.
[20] C. Brzoska, Temporal logic programming and its relation to constraint logic programming, Technical Report, Fak. für Informatik 36/90, University of Karlsruhe, 1990.
[21] C. Brzoska, Temporal logic programming and its relation to constraint logic programming, in: Proc. 1991 Logic Programming Symp., San Diego, California, 1991, MIT Press, Cambridge, MA, pp. 661–677.
[22] C. Brzoska, Temporal logic programming with bounded universal (modality) goals, in: Proc. 10th Internat. Conf. on Logic Programming, Budapest, Hungary, 1993, MIT Press, Cambridge, MA, pp. 239–256.
[23] C. Brzoska, Temporal-logisches Programmieren, Dissertation, Universität Karlsruhe, Fak. für Informatik, 1993.
[24] C. Brzoska, Temporal logic programming in dense time, in: J. Lloyd (Ed.), Proc. 1995 Internat. Logic Programming Symp. Portland, Oregon, 1995, MIT Press, Cambridge, MA, pp. 303–317.
[25] C. Brzoska, Temporal logic programming with metric and past operators, in: M. Fisher, R. Owens (Ed.), Executable Modal and Temporal Logics, Lecture Notes in Computer Science, vol. 897, Springer, Berlin, 1995, Ch. 2, pp. 21–39.
[26] C. Brzoska, K. Schäfer, LIMETTE: Logic programming integrating metric temporal extensions, language definition and user manual, Interner Bericht 9/93, Fak. für Informatik, Universität Karlsruhe, 1993.
[27] C. Brzoska, K. Schäfer, Temporal logic programming applied to image sequence evaluation, in: C. Beierle, L. Plümer (Ed.), Logic Programming: Formal Methods and Practical Applications, Elsevier, Amsterdam, 1995, Ch. 13, pp. 381–395.
[28] C. Brzoska, K. Schäfer, The limette system: Logic programming integrating metric temporal extensions, 1997, Inst. for logic, complexity, and deductive systems, Univ. of Karlsruhe, P.O. Box 6980, D-76 128 Karlsruhe, Germany (to appear in J. Symb. Comp.).
[29] J. Burgess, Basic tense logic, in: D. Gabbay, F. Guenthner (Ed.), Handbook of Philosophical Logic, vol. 2: Extensions of Classical Logic, Reidel, Dordrecht, 1984, pp. 89–133.

[30] H.-J. Bürckert, A Resolution Principle for a Logic with Restricted Quantifiers, Lecture Notes in Artificial Intelligence, vol. 568, Springer, Berlin, 1991.
[31] K.L. Clarke, Negation as failure, in: H. Galliare, J. Minker (Ed.), Logic and Databases, Plenum Press, New York, 1978, pp. 293–322.
[32] J.G. Cleary, V. Kaushik, Updates in a temporal logic programming language, Research Report 91/427/11, Univ. of Calgary, Dept. of Computer Science, 1991.
[33] A. Colmerauer, H. Kanoui, R. Pasero, P. Roussel, Un systéme de communcation homme-machine en francais, Technical report, Université d' Aix-Marseillee, Groupe de Resarche en Intelligence Artificielle, Luminy, 1972.
[34] F. Debart, P. Enjalbert, M. Lescot, Multimodal logic programming using equational and order-sorted logic, Theoret. Comput. Sci. 105 (1992) 141–166.
[35] D. DeGroot, G. Lindstrom (Ed.), Logic Programming, Functions, Relations, and Equations, Prentice-Hall, Englewood Cliffs, NJ, 1986.
[36] L.F. del Cerro, Molog: a system that extends PROLOG with modal logic, New Generation Comput. 4 (1986) 35–50.
[37] L.F. del Cerro, A. Herzig, Automated quantified modal logic, in: P. Brazdil, K. Konolige (Ed.), Machine Learning, Meta-Reasoning and Logics, Kluwer Academic Publishers, Dordrecht, 1990.
[38] N. Dershowitz, Termination of rewriting, J. Symbolic Comput. 3 (1 & 2) (1987) 69–115. Corrigendum: 4 (3) (1987) 409–410.
[39] E.A. Emerson, Modal and temporal logic, in: J. Van Leeuwen (Ed.), Handbook of Theoretical Computer Science, vol. B, Elsevier, Amsterdam and MIT Press, Cambridge, MA, 1990, Ch. 16, pp. 995–1072.
[40] T. Feeß, Ein Tableauverfahren für temporale Hornklausel-Prädikatenlogik, Diplomarbeit, Technische Universität München, Inst. für Informatik, 1991.
[41] M. Fisher, A normal form for first-order temporal formulae, in: D. Kapur (Ed.), Proc. 11th Conf. on Automated Deduction, Saratoga Springs, 1992, Lecture Notes in Artificial Intelligence, vol. 607, Springer, Berlin.
[42] M.J. Fischer, M.O. Rabin, Super-exponential complexity of presburger arithmetic, in: M.R. Karp (Ed.), Complexity of Computations, SIAM-AMS Proc. American Mathematical Society, Providence, RJ, 1974, pp. 27–41.
[43] A.M. Frisch, R.B. Scherl, A general framework for modal deduction, in: J. Allen, R. Fikes, E. Sandewall (Ed.), Proc. Principles of Knowledge Representation and Reasoning, 2nd Internat. Conf. (KR'91), Cambridge, Massachusetts, 1991, Morgan Kaufmann, Los Altos, CA, pp. 196–207.
[44] T. Früwirth, Annotated constraint logic programming applied to temporal reasoning, in: M. Hermenegildo, J. Penjam (Ed.), Proc. 6th Internat. Symp. on Programming Language Implementation and Logic Programming, PLILP '94, Madrid, 1994, Springer, Berlin, pp. 230–243.
[45] M. Fujita, S. Kono, T. Tanaka, T. Moto-oka, Tokio: logic programming language based on temporal logic and its compilation to prolog, in: Proc. 3rd Internat. Conf. on Logic Programming, Lecture Notes in Computer Science, vol. 225, Springer, Berlin, 1986, pp. 695–709.
[46] D.M. Gabbay, Modal and temporal logic programming, in: A. Galton, (Ed.), Temporal Logics and Their Applications, Academic Press, London, 1987, Ch. 6, pp. 197–237.
[47] D.M. Gabbay, Declarative past and imperative future, in: B. Banieqbal, H. Barringer, A. Pnueli (Ed.), Proc. Colloquium on Temporal Logic and Specification, Lecture Notes in Computer Science, vol. 398, Altrincham, 1989, Springer, Berlin, pp. 76–89.
[48] D.M. Gabbay, A temporal logic programming machine, in: T. Dodd, R. Owens, S. Torrance (Ed.), Logic Programming: Expanding the Horizons, Intellect Books, 1991, Ch. 3, pp. 82–123.
[49] D.M. Gabbay, U. Reyle, N-prolog: an extension of prolog with hypothetical implications. I, J. Logic Programming 1 (1984) 319–355.
[50] T. Guckenbiehl, Formalizing and using persistency, in: Proc. 12th Internat. Joint Conf. on Artificial Intelligence, Sydney, Australia, 1991, Morgan Kaufmann, Los Altos, CA.
[51] R. Hale, Temporal logic programming, in: A. Galton (Ed.), Temporal Logics and their Applications, Academic Press, London, 1987, Ch. 3, pp. 91–119.
[52] M. Hanus, The integration of functions into logic programming: From theory to practice, J. Logic Programming 19/20 (1994) 583-628.
[53] E. Harel, O. Lichtenstein, A. Pnueli, Explicit clock temporal logic, in: Proc. 5th Annual IEEE Symp. on Logic in Computer Science, Philadelphia, 1990, IEEE Computer Society Press, Silver Spring, MD.

[54] D. Harel, A. Pnueli, J. Stavi, Propositional dynamic logic of nonregular programs, J. Comput. System Sci. 26 (1983) 222–243.
[55] P.J. Hayes, Computation and deduction, in: Proc. Conf. on Mathematical Foundations of Computer Science, Czechoslovakian Academy of Science, 1973, pp. 105–117.
[56] T. Hrycej, Temporal prolog, in: ECAI 88 Proc. 8th European Conf. on Artificial Intelligence, Munich, 1988.
[57] T. Hrycej, A temporal extension of prolog, J. Logic Programming 15 (1993) 113–145.
[58] G.E. Hughes, M.J. Cresswell, An Introduction to Modal Logic, Methuen, London, 1972.
[59] J. Jaffar, J.-L. Lassez, Constraint logic programming, in: Proc. 14th ACM Symp. on Principles of Programming Languages, Munich, 1986, pp. 111–119.
[60] J. Jaffar, J.-L. Lassez, Constraint logic programming, Technical report, Department of Computer Science, Monash University, Australia, 1986.
[61] J. Jaffar, M.J. Maher, Constraint logic programming: a survey, J. Logic Programming 19/20 (1994) 503–581.
[62] F. Jahanian, A.K. Mook, Safety analysis of timing properties in real-time systems, IEEE Trans. Software Eng. SE-12 (9) (1986) 890–904.
[63] F. Kabanza, J.M. Stevenne, P. Wolpers, Handling infinite temporal data, in: Proc. 9th Annual ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems, Nashville, TN, 1990, pp. 392–403.
[64] J.A. Kamp, Tense Logic and the Theory of Linear Order, Ph.D. Thesis, University of California, Los Angeles, 1968.
[65] P.C. Kanellakis, G.M. Kuper, P.Z. Revesz, Constraint query languages, J. Comput. System Sci. 51 (1995) 26–52.
[66] Y. Kesten, Z. Manna, A. Pnueli, Temporal verification of simulation and refinements, in: J.W. de Bakker, W.-P. de Roever, G. Rozenberg (Eds.), A Decade of Concurrency, Reflexions and Perspectives, Proc. REX School/Symp., Lecture Notes in Computer Science, vol. 803, Springer, Berlin, 1993, pp. 273–346.
[67] R. Kowalski, Predicate logic as a programming language, Inform. Process. 74 (1974, Stockholm, North-Holland) 569–574.
[68] R. Koymans, Specifying Message Passing and Time-Critical Systems with Temporal Logic, Ph.D. Thesis, Technical University of Eindhoven, 1989.
[69] R. Koymans, Specifying real-time properties with metric temporal logic, Real-Time Systems 2 (1990) 255–299.
[70] R. Koymans, J. Vytopil, W.P. de Roever, Real-time programming and asynchronous message passing, in: Proc. 2nd ACM Symp. on Principles of Distributed Computing, Montreal, Canada, 1983, pp. 187–197.
[71] F. Kröger, Logical rules of natural reasoning about programs, in: 3rd Internat. Symp. on Automata, Languages and Programming, Edinburgh University Press, Edinburgh, 1976, pp. 87–98.
[72] F. Kröger, Temporal Logic of Programs, EATCS Monographs on Theoretical Computer Science, vol. 8, Springer, Berlin, 1987.
[73] L. Lamport, Specifying concurrent programs modules, ACM Trans. on Programming Languages and Systems 5 (1983) 190–222.
[74] J.-L. Lassez, M.J. Maher, On fourier's algorithm for linear arithmetic constraints, J. Automat. Reasoning 9 (3) (1992) 373–379.
[75] J.W. Lloyd, Foundations of Logic Programming, Springer, Berlin, 1984.
[76] J. Lobo, J. Minker, A. Rajasekar, Foundations of Disjunctive Logic Programming, MIT Press, Cambridge, MA, 1992.
[77] Z. Manna, A. Pnueli, The Temporal Logic of Reactive and Concurrent Systems: Specification, vol. 1, Springer, Berlin, 1992.
[78] Z. Manna, A. Pnueli, Temporal Verification of Reactive Systems – Safety, vol. 2. Springer, Berlin, 1995.
[79] L.T. McCarty, Clausal intuitionistic logic, I. Fixed-point semantics, J. Logic Programming 5 (1988) 1–31.
[80] L.T. McCarty, Clausal intuitionistic logic, II. Tableau proof procedures, J. Logic Programming 5 (1988) 93–132.
[81] S. Merz, 1990, private correspondence.

[82] S. Merz, Temporal Logic as a Programming Language, Dissertation, Ludwig-Maximilians Universität, München, 1992.
[83] D. Miller, G. Nadathur, F. Pfenning, A. Scedrov, Uniform proofs as a foundation for logic programming, Ann. Pure Appl. Logic 51 (1991) 125–157.
[84] B. Moszkowski, Executing Temporal Logic Programs, Cambridge University Press, Cambridge, 1986.
[85] A. Nonnengart, First-order modal logic theorem proving and functional simulation, in: Proc. 13th Internat. Joint Conf. on Artificial Intelligence, vol. 1, Morgan Kaufmann, Los Altos, CA, 1993, pp. 80–85.
[86] H.J. Ohlbach, A resolution calculus for modal logics, in: E. Lusk, R. Overbeek (Eds.), Proc. 9th Internat. Conf. on Automated Deduction, Lecture Notes in Computer Science, vol. 310, Springer, Berlin, 1988.
[87] H.J. Ohlbach, Semantic-based translation methods for modal logics, J. Logic Comput. 1 (5) (1991) 691–746.
[88] M.A. Orgun, W. Ma, An overview of temporal and modal logic programming, in: D.M. Gabbay, H.-J. Ohlbach (Eds.), Proc. 1st Internat. Conf. on Temporal Logic, ICTL '94, Lecture Notes in Artificial Intelligence, vol. 827, Springer, Berlin, 1994, pp. 445–479.
[89] M.A. Orgun, W.W. Wadge, Towards an unified theory of intensional logic programming, J. Logic Programming 13 (4) (1992) 413–440.
[90] J.S. Ostroff, Temporal Logic of Real-Time Systems, Research Studies Press, Taunton, Somerset, 1990.
[91] A. Pnueli, The temporal logic of programs, in: Proc. 18th Annual Symp. on Foundations of Computer Science, Providence, 1977.
[92] A. Porto, C. Ribeiro. Temporal inference with a point-based interval algebra, in: B. Neumann (Ed.), Proc. ECAI'92 10th European Conf. on Artificial Intelligence, Wien, 1992, Wiley, New York.
[93] V.R. Pratt, The competence/performance dichotonomy in programming, in: Proc. 4th ACM SIGACT/SIGPLAN Symp. on Principles of Programming Languages, California, 1977, pp. 194–200.
[94] A.N. Prior, Past, Present and Future, Clarendon Press, Oxford, 1967.
[95] B. Richards, Y. Jiang, H. Choi, On interval-based temporal planning: An iq strategie, in: Z. Ras (Ed.), Proc. 6th ISMIS'91, Lecture Notes in Artificial Intelligence, vol. 542, Springer, Berlin, 1991.
[96] V. Saraswat, Concurrent Constraint Logic Programming, MIT Press, Cambridge, MA, 1993.
[97] K. Schäfer, Entwicklung einer temporallogischen Sprache zur Beschreibung von Abläufen in Straßenverkehrsszenen, Diplomarbeit, Fak. für Informatik, Universität Karlsruhe (TH), 1993.
[98] P.J. Stuckey, Constructive negation for constraint logic programming, in: Proc. 6th Annual IEEE Symp. on Logic in Computer Science, Amsterdam, 1991.
[99] A. Szalas, Concerning the semantic consequence relation in first-order temporal logic, Theoret. Comput. Sci. 47 (1986) 329–334.
[100] Ch.-S. Tang, Towards a unified logic basis for programming languages, in: R.E.A. Mason (Ed.), Proc. Information Processing 83, Elsevier, Amsterdam, 1983.
[101] A.U. Tanzel, J. Clifford, S. Gadia, S. Jajodia, A Segev, R. Snodgrass (Eds.), Temporal Databases, Theory, Design, and Implementation, Benjamin/Cummings, Menlo Park, CA, 1993.
[102] A. Tuzhilin, J. Clifford, A temporal relational algebra as a basis for temporal relational completeness, in: Proc. Conf. on Very Large Databases, Brisbane, Australia, 1990, pp. 13–23.
[103] J.F.A.K. van Benthem, The Logic of Time, Reidel, Dordrecht, 1983.
[104] J. van Benthem, Correspondence theory, in: D. Gabbay, F. Guenthner (Eds.), Handbook of Philosophical Logic, Extensions of Classical Logic, vol. 2, D. Reidel, Dordrecht, 1984, Ch. II.4, pp. 167–247.
[105] M.H. Van Emden, R.A. Kowalski, The semantics of predicate logic as a programming language, J. ACM 23 (4) (1976) 733–742.
[106] L.A. Wallen, Matrix proof methods for modal logics, in: Proc. 10th Internat. Joint Conf. on Artificial Intelligence, Milan, Italy, 1987, J. Symbolic Comput. 5 (1988) 3–27.
[107] P. Wolper. The tableau method for temporal logic: an overview, Logique et Analyse 28 (1985) 119–136.